



HAL
open science

Systèmes neuromorphiques: Etude et implantation de fonctions d'apprentissage et de plasticité

Adel Daouzli

► **To cite this version:**

Adel Daouzli. Systèmes neuromorphiques: Etude et implantation de fonctions d'apprentissage et de plasticité. Micro et nanotechnologies/Microélectronique. Université Sciences et Technologies - Bordeaux I, 2009. Français. NNT: . tel-00573417

HAL Id: tel-00573417

<https://theses.hal.science/tel-00573417>

Submitted on 3 Mar 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

présentée à

L'UNIVERSITÉ DE BORDEAUX I

ÉCOLE DOCTORALE DE SCIENCES PHYSIQUES ET DE L'INGENIEUR

par **Adel Mohamed DAOUZI**

POUR OBTENIR LE GRADE DE

DOCTEUR

SPÉCIALITÉ : ELECTRONIQUE

**Systemes neuromorphiques: Etude et implantation de
fonctions d'apprentissage et de plasticité.**

Soutenance le : 18 juin 2009

Après avis des rapporteurs :

M.	Stéphane BINCZAK	Maître de Conférences (HDR)	Rapporteur
M.	Alain FAURE	Professeur	Rapporteur

Devant la commission d'examen composée de :

M.	Stéphane BINCZAK	Maître de Conférences (HDR)	Rapporteur
M.	Alain FAURE	Professeur	Rapporteur
Mme	Sylvie RENAUD	Professeur	Directeur de thèse
M.	Sylvain SAÏGHI	Maître de Conférences	Co-directeur de thèse
M.	Jacques HENRY	Directeur de Recherche INRIA	Président du jury
M.	Andrew DAVISON	Chargé de Recherche CNRS	

THÈSE

présentée à

L'UNIVERSITÉ DE BORDEAUX I

ÉCOLE DOCTORALE DE SCIENCES PHYSIQUES ET DE L'INGENIEUR

par **Adel Mohamed DAOUZI**

POUR OBTENIR LE GRADE DE

DOCTEUR

SPÉCIALITÉ : ELECTRONIQUE

**Systemes neuromorphiques: Etude et implantation de
fonctions d'apprentissage et de plasticité.**

Soutenance le : 18 juin 2009

Après avis des rapporteurs :

M.	Stéphane BINCZAK	Maître de Conférences (HDR)	Rapporteur
M.	Alain FAURE	Professeur	Rapporteur

Devant la commission d'examen composée de :

M.	Stéphane BINCZAK	Maître de Conférences (HDR)	Rapporteur
M.	Alain FAURE	Professeur	Rapporteur
Mme	Sylvie RENAUD	Professeur	Directeur de thèse
M.	Sylvain SAÏGHI	Maître de Conférences	Co-directeur de thèse
M.	Jacques HENRY	Directeur de Recherche INRIA	Président du jury
M.	Andrew DAVISON	Chargé de Recherche CNRS	

Remerciements

Je tiens avant tout à remercier très cordialement le professeur Sylvie Renaud de m'avoir offert l'opportunité de réaliser ces travaux de thèse, et le Maître de Conférence Sylvain Saïghi de m'avoir encadré tout au long.

Je remercie également Alain Destexhe et Michelle Rudolph pour leurs conseils scientifiques enrichissants, et la communauté FACETS en général.

Je remercie le professeur Alain Faure et le Maître de Conférence Stéphane Binczak d'avoir accepté d'être mes rapporteurs et de m'avoir apporté des remarques et suggestions fort intéressantes.

Un grand merci aussi au directeur de recherche Jacques Henry et au chercheur Andrew Davison pour leurs suggestions et pour avoir participé à mon jury de thèse.

Je remercie également toute l'équipe Ingénierie des Systèmes Neuromorphiques pour les diverses choses qu'elle m'a apporté et notamment les traditionnels goûters. Plus particulièrement un grand merci à Olivia pour son amitié, Gilles pour son humour, Guilherme pour son agréable humeur, Colin pour les bons débuts, Jean pour la group attitude, Noëlle pour sa gentillesse, Yannick et Bilel pour leur collaboration...

Je tiens à remercier particulièrement d'autres collègues du laboratoire, et pour certains amis, qui m'ont chacun marqué à leur manière, je cite entre autres Benjamin, Arnaud, François, Ludo, Marie-Lise, Chacha, Willy, Seb, Johnny, Timo, Hakim, Fabien, Yann, et bien d'autres.

Je tiens aussi à remercier mes amis, notamment Seb D., Manu, Fanny, les Cléments, Guitou, Benjamin & Fix, Jérémie, Chloé, Elsa, Sarah, mes voisins,

et j'en passe des meilleurs. Et une pensée particulière pour Sylvain P., Truch, Jim, Gus, Jul, Polak, Yo, Lionel, Brice et Aurélia ;)

Je garde le meilleur pour la fin avec bien entendu ma famille qui compte énormément pour moi et ma douce Céline qui m'a tellement aidé du début à la fin de cette aventure.

Enfin, je n'oublierai pas, bien évidemment, toute la communauté du libre qui m'a apporté énormément pour mes travaux passés, présents et futurs. Merci à tous ceux qui mettent en commun leurs connaissances et leurs savoirs et contribuent à l'amélioration de nos richesses intellectuelles universelles. En parlant d'intellect et d'universalité, j'aimerais remercier pour son génie Noam Chomsky l'un des plus grands intellectuels toujours vivant !

Résumé

Dans ces travaux de thèse, nous nous sommes intéressés à l'influence du bruit synaptique sur la plasticité synaptique dans un réseau de neurones biophysiquement réalistes. Le simulateur utilisé est un système électronique neuromorphique. Nous avons implanté un modèle de neurones à conductances basé sur le formalisme de Hodgkin et Huxley, et un modèle biophysique de plasticité. Ces travaux ont inclus la configuration du système, le développement d'outils pour l'exploiter, son utilisation ainsi que la mise en place d'une plateforme le rendant accessible à la communauté scientifique via Internet et l'utilisation de scripts PyNN (langage de description de simulations en neurosciences computationnelles).

Mots-clés

Systèmes Neuromorphiques, Hodgkin et Huxley, Réseaux de neurones, Plasticité, STDP, PyNN, Neurosciences Computationnelles

Abstract

In this work, we have investigated the effect of input noise patterns on synaptic plasticity applied to a neural network. The study was realised using a neuromorphic hardware simulation system. We have implemented a neural conductance model based on Hodgkin and Huxley formalism, and a biophysical model for plasticity. The tasks performed during this thesis project included the configuration of the system, the development of software tools, the analysis tools to explore experimental results, and the development of the software modules for the remote access to the system via Internet using PyNN scripts (PyNN is a neural network description language commonly used in computational neurosciences).

Keywords

Neuromorphic Systems, Hodgkin & Huxley model, Neural Networks, Plasticity, STDP, PyNN, Computational Neurosciences

Table des matières

Introduction	i
1 Neurosciences et neuromorphisme	1
1.1 Etat de l'Art et contexte scientifique	1
1.1.1 Associativité/Corrélation	1
1.1.2 De la découverte du neurone à la plasticité	4
1.1.3 Modélisation	6
1.1.4 Conclusion	12
1.2 Environnement des travaux de thèse	12
1.2.1 Présentation de l'équipe "Ingénierie des Systèmes Neuromorphiques"	12
1.2.2 Projet pluridisciplinaire FACETS et historique	14
1.2.3 Modélisation	16
1.2.4 conclusion	17
1.3 Modèles	17
1.3.1 Physiologie du Neurone	17
1.3.2 Synapses	23
1.3.3 Plasticité (STDP)	27
1.3.4 Bruit	31
1.3.5 Conclusion	32
1.4 Conclusion	32
2 Bruit corrélé et plasticité	35
2.1 Système de simulation électronique PAX	36
2.1.1 Propriétés du système	36
2.1.2 Automatisation du système	42
2.1.3 Détermination des valeurs biophysiques manipulées	44

2.2	Préparation de l'étude et de l'analyse	47
2.2.1	Corrélation des patterns de bruit	48
2.2.2	Outils d'analyse	49
2.3	Effet de la corrélation du bruit sur la plasticité	55
2.3.1	Corrélation identique des neurones	55
2.3.2	Corrélation dans 2 groupes de neurones	64
2.3.3	Conclusion	70
3	Environnement neuromorphique Gaillimh	73
3.1	Système de simulation électronique Gaillimh	73
3.1.1	Propriétés du système et plasticité	73
3.1.2	Réglage des circuits - Méthodes de réglage	80
3.2	Utilisation du système	93
3.2.1	Propriétés fonctionnelles des neurones	93
3.2.2	Simulation avec Gaillimh	95
3.3	Mise à disposition du système	100
3.3.1	Développement avec PyNN	100
3.3.2	Accès au système	104
3.3.3	Conclusion	105
	Discussion et conclusion	107
	Publications de l'auteur	115
	Bibliographie	117
	A Paramètres du modèle HH	125
	B Algorithme itératif de STDP	127
	C Formats, programmes et scripts	129
	C.0.4 Formats des données	129
	C.0.5 Programmes et scripts	136
	D Protocole de communication de Gaillimh	141
	D.0.6 Attribution des ports série	141
	D.0.7 Paramètres analogiques et numériques	141
	D.0.8 Gestion du bruit synaptique	143
	D.0.9 Réception des données	145

Introduction

"Je pense que le XXI^e siècle sera celui du cerveau, comme le XX^e a été celui de l'atome et de la biologie moléculaire"

J.-P. CHANGEUX

Le système nerveux livre de plus en plus de ses secrets de fonctionnement. Nous avons appris à l'appréhender à différentes échelles, allant de l'étude du comportement et de la cognition humaine et animale, en passant par l'étude de structures cérébrales et de réseaux de neurones et jusqu'aux composants même de ces cellules nerveuses. Des disciplines diverses se sont retrouvées pour saisir les différents aspects qui forment notre connaissance des sciences du cerveau. La psychologie amène la neuropsychologie et les neurosciences en se mêlant à la biologie. La chimie amène la biochimie. Les mathématiques amènent la modélisation et les sciences de la technologie la simulation de ces modèles. En combinant tous ces domaines, tous ces savoirs, la complexité du cerveau humain devient un terrain d'exploration débordant de ressources.

Depuis des années nous savons que la synchronisation de plusieurs régions du cerveau est liée à la mise en commun de ressources dédiées à certains traitements cognitifs. Le système nerveux permet aux hommes et aux animaux une incroyable adaptation à leur environnement changeant. Cela leur permet par exemple de combiner des informations sensorielles différentes (audition, vision, odorat,..) afin de prendre très rapidement des décisions et d'adopter un comportement qui va de la survie aux plaisirs. Des insectes, au système nerveux rudimentaire, ont la capacité de se mouvoir dans un environnement riche en stimuli et de s'adapter à différentes situations avec une grande réactivité. Les cellules nerveuses présentent une activité électrique dont les signaux atteignent des fréquences de l'ordre de quelques dizaines, voire de quelques centaines de Hertz. Alors que la performance de nos technologies s'accroît en termes de fréquence d'exécution (de nos jours de l'ordre du Giga-

Herz), de capacité de stockage d'informations, il n'en demeure pas moins que nous sommes confrontés à des limitations dans la capacité de traitement d'informations lorsque celles-ci sont nombreuses, imprévisibles et demandent une réponse rapide. Malgré la rapidité de traitement des systèmes de plus en plus sophistiqués, nous n'arrivons pas à obtenir des systèmes artificiels qui auraient des capacités de traitements cognitifs équivalentes à celles de systèmes nerveux assez primitifs. Cela amène la communauté scientifique à reconsidérer son approche du calcul et remettre en question les performances permises par des systèmes hérités du concept de la « machine de Turing » et utilisant des algorithmes pourtant de plus en plus optimisés et performants. Ainsi, l'étude des systèmes biologiques est plus que jamais d'actualité et les scientifiques, en tentant de percer les mystères des systèmes nerveux s'en inspirent pour trouver de nouveaux paradigmes de calcul.

C'est ainsi que les domaines de recherche dans le champs de la modélisation et de la simulation des systèmes nerveux connaissent une très forte activité. De nombreux scientifiques, notamment de neurophysiologistes, se sont attelés ces dernières années à développer et à préciser les modèles de neurones, de synapse et de réseaux de neurones. L'avènement de l'informatique et de l'électronique permet de simuler avec différents niveaux de précision les mécanismes nerveux.

Les travaux réalisés dans le cadre de cette thèse s'inscrivent dans le contexte de développement d'outils dont le fonctionnement s'inspire du biologique.

Dans le premier chapitre nous aborderons des notions liées à nos connaissances générales sur le système nerveux. Puis nous dresserons un état de l'art des outils de simulation existant actuellement. Nous approfondirons ensuite les notions utiles à la compréhension des mécanismes sous-jacents aux propriétés des systèmes neuromorphiques que nous utilisons dans nos travaux ainsi qu'aux propriétés que nous explorons.

Dans le second chapitre, nous étudierons l'effet de la corrélation des entrées de neurones d'un petit réseau simulé et soumis à des mécanismes de plasticité. Nous avons réalisé cette étude avec un système neuromorphique dont nous présenterons les principales caractéristiques et le mode d'utilisation.

Dans le troisième chapitre, nous présenterons un système neuromorphique possédant plus de degrés de liberté pour son réglage. Ainsi nous ver-

rons comment nous pouvons configurer ce système, comment nous pouvons implanter un modèle de neurone particulier et quelles sont les propriétés de ces neurones. Ce système permet également de calculer la plasticité automatiquement. Nous montrerons ensuite l'utilisation faite du système configuré. Pour terminer, nous présenterons les travaux réalisés afin de développer des plateformes permettant de rendre accessibles, via le réseau Internet, ces systèmes neuromorphiques à la communauté scientifique.

Pour terminer nous discuterons des travaux réalisés dans cette thèse et reprendrons les apports des systèmes neuromorphiques au domaine de la simulation de systèmes nerveux, et leurs perspectives.

Chapitre 1

Des neurosciences et des systèmes neuromorphiques

Dans ce chapitre nous allons évoquer des points importants de nos connaissances concernant le système nerveux et plus particulièrement sa modélisation. Nous dresserons également un état de l'art des solutions de simulation. Puis nous aborderons le contexte plus général de ces travaux de thèse. Et enfin nous verrons les notions importantes nécessaires à la compréhension des outils et des études réalisées.

1.1 Etat de l'Art et contexte scientifique

1.1.1 Associativité/Corrélation

Les techniques récentes d'enregistrements électrophysiologiques, électromagnétiques, d'imagerie fonctionnelle et de simulation nous permettent de sonder plus en avant les secrets du fonctionnement du cerveau. Ce qui est en général étudié est l'activité d'un grand nombre de cellules nerveuses activées simultanément. Lorsque dans certaines études, en IRMf¹ ou utilisant des enregistrements EEG², on observe plusieurs régions activées, il s'agit en fait de différentes populations de neurones qui ont leur activité plus ou moins corrélée temporellement et/ou spatialement. Le propre des neurones est de pouvoir modifier leurs affinités de communication afin d'être en mesure de synchroniser leur activité.

¹Imagerie par Résonance Magnétique fonctionnelle

²Electro-Encéphalogramme

On s'accorde souvent à présenter les mécanismes de mémoire et d'apprentissage comme étant la grande richesse des systèmes nerveux évolués, trouvant son apogée dans le cerveau humain. Ainsi le cerveau permet d'apprendre et de restituer le fruit de l'apprentissage. Il permet de s'adapter à un environnement changeant. Il apporte la capacité d'associativité, fer de lance de l'adaptation. Cette capacité passe par les mécanismes de synchronisation et de corrélation d'activité, le tout sous-tendu par les mécanismes dits de plasticité.

- Associativité

La notion d'associativité peut être reliée à celle de mémoire associative. Le concept cognitif de mémoire associative peut être décrit par l'association d'entités de même nature ou pas, liées au cours d'un événement répété ayant impliqué des séquences d'activations particuliers (ou *patterns*). Il est possible d'extrapoler cette notion à l'association d'idées, de comportements face à une situation, à toute autre situation issue d'un apprentissage. Par exemple lorsque nous pensons à une personne, nous lui associons un timbre vocal, un visage, une physionomie et d'autres éléments qui lui sont particuliers. Ainsi, à force de rencontrer cette personne nous lui avons associé un ensemble de propriétés qui la caractérisent. Le fait d'accéder à un sous-ensemble de ces caractéristiques réactive les autres propriétés associées. Pour revenir à l'associativité, on pourrait dire que le fait d'avoir une représentation composée de plusieurs éléments, implique que l'accès à un ensemble d'éléments entraîne la réactivation du tout. Ainsi, pour l'appliquer au système nerveux, considérons de vastes populations de neurones qui sont activées simultanément à plusieurs reprises dans le cas de la soumission à un ensemble d'informations perceptives particulières. Par exemple, le processus cognitif de l'identification d'une personne en passant par sa représentation visuelle et auditive. A force d'avoir les deux informations en même temps, les réseaux de neurones permettant cette identification développent des liens permettant par la suite qu'une seule activation réactive la seconde.

- Synchronisation et corrélation

Pour l'étude du traitement de l'information par le cerveau, initialement l'activité des neurones était enregistrée unitairement, ce qui signifie que l'on s'intéressait à l'activité d'un seul neurone. Ceci a mis en exergue le rôle important de la fréquence de décharge. Avec l'évolution des techniques d'enregistrements simultanés, on a commencé à s'intéresser à l'incidence des mes-

sages nerveux (communément appelés *Potentiels d'Action* ou PA) entre différents neurones qui présentent alors dans certains cas une activité synchronisée [1]. Singer et Gray proposent en 1995 [2] qu'une corrélation des PAs de différents neurones devrait avoir lieu dans une même colonne corticale pour le codage de propriétés locales. Une telle corrélation devrait également avoir lieu entre différentes colonnes d'une même aire pour le traitement de propriétés liées mais réparties spatialement. De même, une corrélation entre différentes aires corticales permettrait de mettre en relation les informations de différentes catégories et des localisations différentes dans l'espace. D'après les auteurs, il existerait également une corrélation entre les deux hémisphères cérébraux pour lier l'information des deux champs visuels et également une corrélation entre des modalités sensorielles et motrices pour l'intégration des traitements sensorimoteurs. Ils s'appuient sur des évidences anatomiques et physiologiques pour soutenir que le traitement d'images visuelles s'effectue dans différentes parties du cortex visuel, et que la représentation d'objets implique une action concertée de larges populations de neurones distribuées à travers le cortex visuel. Ils précisent que l'identification de propriétés liées d'un même objet perçu est portée par la coïncidence temporelle de décharges neuronales induites par ces propriétés. De plus, dans de larges réseaux de neurones, les cellules reçoivent des entrées convergentes depuis plusieurs milliers de différentes cellules. Ces suppositions semblent vérifiées par des travaux plus contemporains. On a observé dans différentes structures cérébrales une corrélation des PAs entre différents neurones. On observe dans la rétine (chat, lapin) des PAs synchronisés à différents niveaux qui seraient liés au partage d'entrées synaptiques des champs récepteur des cellules ganglionnaires. Cette synchronisation aurait un rôle d'amplification du signal de faibles stimuli visuels [3]. Cette synchronisation se retrouve également à l'étape de traitement visuel suivante, c'est-à-dire au niveau du corps géniculé latéral (LGN³) où 20 à 40 % des PAs de neurones voisins, dont les champs récepteurs se chevauchent, ont une différence temporelle de l'ordre de la milliseconde. Une synchronisation des neurones d'une même aire corticale mais des hémisphères opposés a pu être observée [4]. On observe également une synchronisation de neurones appartenant à des aires corticales différentes, comme par exemple dans les aires V1 et V2 du cortex visuel du même hémisphère du cerveau d'un macaque [5]. On observe une synchronisation des

³Lateral Geniculate Nucleus

PAs, durant de courtes périodes (100-200 ms) en phase terminale de préparation du mouvement dans le cortex moteur de singes. Il a été montré non seulement que la fréquence de décharge des neurones codait de l'information mais que la coïncidence des PAs également. Des études montrent que la précision temporelle de coïncidence des PAs dans des fonctions cognitives de haut niveau serait comprise entre 1 et 5 ms [6, 7]. D'après Kopell [8], avec l'enregistrement de potentiels évoqués (activité électrique en réponse à des stimulations), on observe des synchronisations sur de longues distances avec des rythmes bêta (12-30 Hz), alors qu'on observe des synchronisations locales dans une bande gamma (30-80 Hz). Ces bandes de fréquences jouent un rôle dans les processus de cognition, d'attention, de perception. On observe également que les rythmes alpha peuvent apparaître dans le cadre d'une désynchronisation active sur la distance avec un effet de synchronisation ou non localement. Nous sommes capables de segmenter une scène et en distinguer différents objets indépendants en une fraction de secondes.

1.1.2 De la découverte du neurone à la plasticité

La technique de coloration permettant de voir les neurones, découverte par Golgi, et son utilisation dans les travaux de Ramon y Cajal [9], à la fin du XIXe siècle, ont conduit à la représentation du neurone en tant qu'entité cellulaire anatomique et fonctionnelle. Ainsi est née la « théorie du neurone », finalisée et ainsi nommée par Wilhelm von Waldeyer. Cependant Golgi et Ramon y Cajal proposaient des représentations différentes du réseau de neurones. Pour le premier, les neurones étaient constitués en réseau continu, comme le réseau de veines par exemple. Pour le second une discontinuité existait entre les neurones. Le développement de la microscopie électronique dans les années 1950 donnera définitivement raison à Cajal. En effet, la microscopie optique ordinaire possède une résolution limite de $0,1 \mu m$, or l'espace entre deux neurones n'est que de $0,02 \mu m$ ce qui ne permettait pas de trancher entre les conclusions opposées de Golgi et Cajal jusqu'à ce que la limite de $0,1 nm$ du microscope électronique permette de résoudre le problème. Les neurones, indépendants les uns des autres, sont donc organisés en réseaux, et des impulsions électriques parcourent ces cellules. Au début du XXe siècle, Charles Scott Sherrington introduit le terme de « synapse » pour désigner la zone de discontinuité entre deux neurones suggérée par Ramon y Cajal. Ce n'est qu'en 1949 que Donal Hebb [10] formule la première hypo-

thèse d'une plasticité qui modifierait l'efficacité synaptique. Le postulat de Hebb s'énonce comme suit :

« Quand un axone de la cellule A est assez proche pour exciter une cellule B et quand, de façon répétée et persistante, il participe à son activation, un certain processus de croissance ou un changement métabolique s'installe dans une cellule ou dans les deux tel que l'efficacité de A, en tant qu'elle est une des cellules qui active B, est augmentée. »

Donald HEBB

Il ajoute également que « concernant la façon dont une cellule pourrait augmenter sa capacité à en activer une autre est de dire que les boutons synaptiques développent et augmentent la surface de contact entre l'axone afférent et le soma efférent ("soma" réfère aux dendrites et au corps cellulaire ou à toute autre cellule excepté son axone). »

Cette formulation constitue la première règle d'apprentissage énoncée. Elle a été largement utilisée dans le domaine des neurosciences computationnelles. Les modèles de plasticité décrivant les mécanismes d'apprentissage, peuvent être appliqués à des fonctions de catégorisation et de classification. Plusieurs modes d'apprentissage existent. Citons par exemple l'apprentissage supervisé et non-supervisé. Dans le premier cas, un superviseur contraint l'état interne du réseau de neurones afin d'obtenir des sorties particulières pour des entrées données par une phase d'apprentissage. Dans l'autre cas, l'état interne du réseau de neurone change en fonction des entrées qu'il reçoit, et l'organisation ne dépend pas alors d'états de sortie imposés. Dans le système nerveux central, la plasticité contribue au développement, à l'apprentissage et la mémoire, et permet la récupération en cas de lésion. Des neuropathologies sont liées à la plasticité, telles que l'épilepsie, les neurodégénérescences et la dépendance aux drogues. Le postulat de Hebb a progressivement laissé la place à des explications physiologiques plus précises. Ainsi les bases de la plasticité, liée à l'apprentissage et à la mémorisation, seraient explicables par les mécanismes de potentialisation à long terme (LTP⁴) et de dépression à long terme (LTD⁵) [11, 12, 13]. Les notions de LTP et LTD suggèrent que l'efficacité de la connexion synaptique entre deux neurones peut être renforcée et diminuée respectivement. Ainsi la LTP correspondrait à l'aug-

⁴Long-Term Potentiation

⁵Long-Term Depression

mentation de l'efficacité postulée par Hebb, la LTD vient compléter son postulat. Des stimulations magnétiques du cortex somatosensoriel de l'humain suggèrent également l'existence d'une plasticité synaptique dépendant du temps d'émission de PA, connue sous le nom de STDP⁶. Ce genre de règle de plasticité a été élaboré à partir d'expérimentations en neurophysiologie sur lesquelles nous reviendrons plus en détail ultérieurement.

1.1.3 Modélisation

Depuis la découverte du neurone au début du XXe siècle, sa caractérisation n'a cessé d'évoluer jusqu'à permettre une étude plus poussée, avec l'avènement de l'informatique, de sa modélisation et l'utilisation de ces modèles pour le calcul et les neurosciences. Si au début, les modèles étaient produits par des neuroscientifiques (neurologues, une partie des psychologues, physiologistes), la modélisation de réseaux de neurones s'est petit à petit éloignée de la considération biologique pour devenir un outil en mathématiques ou en physique [14].

Modèles connexionnistes, dits neuro-mimétiques.

La modélisation de la mémoire supposait, d'après les travaux de Lashley dès 1950 [15], une localisation commune pour des sous-types de ce que l'on nommait mémoire. Ainsi, des structures communes auraient différents modes de fonctionnement se rattachant à différents types de mémoires. La modélisation connexionniste a abouti à la modélisation de réseaux d'unités élémentaires appelées neurones formels (figure 1.1) dont la première célèbre description a été apportée par les travaux de McCulloch et Pitts en 1943 [16]. La fonction d'activation qui détermine l'état du neurone est à l'origine une fonction Heaviside, mais a été souvent remplacée par la suite par une fonction sigmoïde ou une fonction tangente hyperbolique.

⁶Spike Timing-Dependent Plasticity

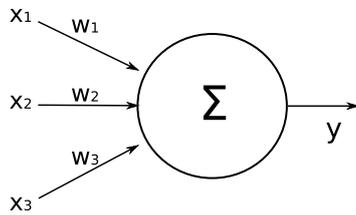


FIG. 1.1 – Exemple de neurone formel à 3 entrées. La sortie est le résultat binaire de la comparaison d'une somme pondérée des entrées à un certain seuil

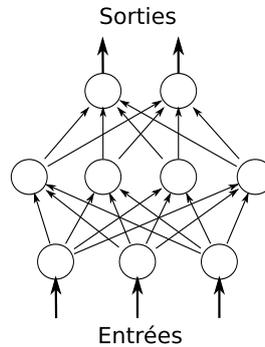


FIG. 1.2 – Exemple de Perceptron à multi-couches

Puis les modèles se complexifiant, les capacités de calcul évoluèrent avec notamment le Perceptron [17] (réseau de neurones simple constituant un classifieur linéaire), appliqué par exemple à la reconnaissance de formes. Ce réseau de neurones, incapable de résoudre des problèmes non-linéaires [18], a été amélioré pour donner en 1986 le MLP⁷ [19], appelé en français perceptron à multi-couches. L'exemple typique du MLP possède trois couches de cellules (une couche d'entrée, une couche cachée et une couche de sortie) comme le montre la figure 1.2. La couche d'entrée peut être considérée en tant que couche sensorielle. La couche de sortie peut correspondre à une couche motrice. La couche cachée serait alors une couche abstraite pouvant faire office de couche de représentation, de mémorisation. Un autre modèle de réseaux de neurones qui permet de lever les limitations concernant les problèmes non-linéaires, est le réseau de Hopfield, modèle de réseau récurrent constituant la première application d'une mémoire associative [14]. La modélisation des neurones a été enrichie de règles de plasticité pour les faire fonctionner en réseau et leur permettre d'intégrer des règles d'apprentissage. La plus célèbre règle de plasticité, inspirée du vivant, citée précédemment, nous a été fournie par le psychologue américain Donald Hebb en 1949 [10]. Plus tard, notamment pour régler le Perceptron, d'autres règles, plus mathématiques (comme l'apprentissage par rétropropagation de l'erreur dans le cas du MLP), ont permis d'adapter les réseaux de neurones à l'univers du calcul et de la computation. Précisons que la capacité d'adaptation des modèles de réseaux de neurones réside dans les algorithmes de plasticité synaptique [20].

⁷ *Multi-Layer Perceptron*

L'avantage des réseaux connexionnistes est leur capacité d'apprentissage et d'adaptation. Le réseau possède une mémoire et peut accumuler des connaissances. La modélisation a évolué principalement vers deux voies que sont le calcul, et la représentation du vivant. D'un côté les réseaux de neurones artificiels permettent d'effectuer de la catégorisation et pallier les limitations des algorithmes plus classiques, et de l'autre ils permettent d'explorer plus en avant les propriétés fonctionnelles et structurelles des systèmes nerveux et parfois rendre compte de leurs mécanismes. Dans un souci de réalisme biologique, de dynamique et de performances de calcul, des modèles de neurones intégrant la dimension temporelle, ou fréquentielle, absentes des premiers modèles de neurones ont commencé à supplanter les modèles formels. Ces modèles, connus sous l'expression anglophone de "*spiking neurons*", sont couramment utilisés de nos jours. La plupart de réseaux de neurones de type Hopfield ou à fonction sigmoïdale peuvent être émulsés en codage temporel [21]. Ils permettent de réaliser de façon plus performante les mêmes tâches que les modèles de neurones formels [22]. Il existe plusieurs familles de *spiking neurons*, citons par exemple les modèles de type *Integrate and Fire* (I&F) à fuite ou non-linéaires [23]. Ils rendent compte des temps d'émission de PAs mais sans reproduire leur forme. Les modèles SRM⁸ apportent en plus la forme du PA. Les *spiking neurons* proposés par Izhikevich permettent de rendre compte non seulement de la forme du PA mais en plus de différents types d'activité de divers neurones [24]. Les modèles de *spiking neurons* sont aussi bien utilisés à des fins de computation que de modélisation réaliste. Avec la significative amélioration des capacités de calcul des ordinateurs ces dernières décennies, et la grandissante contribution de l'électronique pour la simulation de réseaux de neurones, des modèles plus anciens et plus précis, basés sur la dynamique des canaux ioniques transmembranaires, sont revistés et implantés. Ajoutons enfin que la précision et la pertinence biologique dépendent de la complexité du modèle simulé.

Environnements de simulation informatique

Il existe un grand nombre d'outils de simulation informatique, nous présentons ici les plus connus dans la communauté de modélisation en neurosciences. Une revue des différents simulateurs logiciels est proposée par Brette

⁸*Spike Response Model*

et al. [25].

-Logiciel NEURON :

L'une des plate-formes les plus répandues est sans aucun doute le logiciel libre NEURON [26] permettant de simuler un grand nombre de modèles biophysiques pouvant se baser sur une architecture spatiale et une dynamique temporelle, et sur le formalisme de type Hodgkin et Huxley [27] modélisant un ou plusieurs compartiments à conductances. Ce très puissant outil a été utilisé dans des centaines d'études différentes, allant de l'étude des propriétés de certaines substances chimiques d'une membrane, à l'étude d'interactions dans de larges réseaux de neurones. Les traitements peuvent être parallélisés sur des systèmes prévus à cet effet.

-Logiciel GENESIS :

GENESIS est un autre outil libre de simulation des systèmes neuronaux et biologiques. Il permet de simuler des réactions biochimiques et composants subcellulaires, des modèles complexes de neurone, mais aussi de grands réseaux de neurones. A l'instar de NEURON, il permet également, grâce à l'extension PGENESIS, d'être utilisé avec des systèmes fonctionnant de manière parallélisée.

-Logiciel NEST :

NEST est un projet collaboratif libre. Il a pour spécialité la simulation de grands réseaux de neurones à la connectivité biologiquement réaliste. Les réseaux simulés peuvent atteindre la taille de 10^5 neurones.

-Logiciel NCS :

Le *NeoCortical Simulator* (NCS) est optimisé pour simuler des structures de type corticales. Il respecte la structure 3D des neurones, et intègre également un calcul de plasticité. Il peut également fonctionner sur des systèmes parallélisables.

-Projet *Blue Brain*

Les programmes implantés sur *Blue Gene* sont développés en langage C/C++ en utilisant le modèle de programmation standard MPI2 (*Message Passing Interface 2*). Ce projet [28], fruit d'une collaboration entre IBM et l'EPFL⁹, a pour vocation de modéliser à un niveau très détaillé le cortex cérébral en utilisant le supercalculateur *Blue Gene* d'IBM. Un problème qu'un seul processeur pourrait résoudre en 57 jours, la machine qui compte 8 192 processeurs peut le faire en 10 minutes. Une des applications effectuée

⁹Ecole Polytechnique Fédérale de Lausanne

dans le cadre du projet *Blue Brain* a été de simuler de manière efficace un modèle de 1 mm^3 de tissu du cerveau d'un rat.

Environnements de simulation matérielle

Malgré la parallélisation des processus de traitement des logiciels suscités, les performances de calcul des ordinateurs impliquent des temps de simulation qui peuvent être particulièrement importants sur des systèmes légers. Ainsi, des systèmes matériels dédiés commencent à apparaître. C'est en 1991 que l'équipe de Carver Mead propose une solution d'intégration analogique sur silicium pour simuler un modèle biophysique de neurone basé sur le formalisme de Hodgkin et Huxley [29]. Ajoutons que dans beaucoup de cas, l'intérêt principal est de communiquer avec le vivant. Les circuits électroniques simulant des réseaux de neurones sont appelés systèmes neuromorphiques ou encore neuromimétiques car ils miment le fonctionnement des neurones.

-Système de Shimonomura

Ce système neuromorphique s'inspire de la structure du cortex visuel primaire afin de réaliser un traitement automatique d'informations visuelles en temps réel [30]. Dans ce système les neurones sont simulés de façon comportementale. Il est constitué d'une rétine artificielle sur silicium simulant les circuits de neurones rétiniens. Puis une carte de sélectivité d'orientation reproduit l'effet d'un filtre de Gabor. Enfin un circuit numérique programmable (FPGA¹⁰) effectue les traitements de cellules plus complexes.

-Système de Indiveri

Le système présenté par Indiveri et Fusi [31] se base sur un modèle de type I&F et fournit des réseaux comptant un petit nombre de neurones. Sur une surface de $6,1\text{ mm}^2$, avec une technologie CMOS $0,35\mu\text{m}$, le circuit fournit 16 neurones avec 128 synapses possibles par neurone. Un calcul de plasticité, dont les synapses ne peuvent avoir que 2 états stables, est incorporé au système électronique et permet d'effectuer des classifications après un apprentissage semi-supervisé [32].

-Système de Wijekoon

Le système décrit dans [33] présente une puce VLSI contenant 202 neurones qui peuvent simuler 4 types de cellules corticales différentes. Ces tra-

¹⁰Field-Programmable Gate Array

voux ont pour vocation de reproduire différents types d'activités des neurones corticaux. La technologie utilisée est CMOS 0,35 μ m.

-Système de Hynna

Leur système présenté dans [34] permet de simuler des neurones relais sensoriels au niveau du thalamus produisant une activité par *burst* (série de quelques PAs à haute fréquence > 250 Hz) ou par des PAs avec une plus faible fréquence (< 100Hz). Ici l'intérêt est plus particulièrement porté sur le fait de reproduire une activité déterminée. Une puce de 11mm² en technologie CMOS 0,25 μ m peut fournir 7200 neurones.

-Systèmes électroniques de Schemmel

L'équipe de Karlheinz Meier au *Kirchoff-Institut für Physik* a développé le système HAGEN [35], composé de circuits analogiques VLSI¹¹, permettant de simuler des réseaux d'environ 4k neurones et 512 synapses de type Mc Culloch et Pitts appliqués au traitement de l'image. La vitesse de fonctionnement est largement supérieure au temps réel biologique (de l'ordre de 10⁴).

Leur nouveau système de simulation matérielle en cours de développement dans le cadre du projet européen FACETS est directement implanté sur des *wafers* (galettes de silicium) [36]. Cela permet de garder des performances substantielles et un temps de traitement encore supérieur au temps réel biologique simulé. Un *wafer* de 20 cm de diamètre peut contenir jusqu'à 60 millions de synapses. Cette fois-ci le modèle de neurone utilisé est de type I&F avec fuite. La connectivité synaptique, bien que pouvant avoir une forte densité, ne peut être totale (*all-to-all* en anglais). Le système permet également de simuler une plasticité synaptique [37].

-Système de Renaud

Le système développé par notre équipe [38] fournit des *spiking neurons* dont le modèle de neurone est basé sur le formalisme de Hodgkin et Huxley (*conductance-based*). La simulation des neurones est à la charge de circuits analogiques, la gestion du réseau incombe à un circuit FPGA, le tout est intégré sur une carte électronique.

La première version du système « PAX » permet uniquement la simulation de réseau de neurones alors que le calcul de plasticité est à la charge d'un ordinateur. La technologie utilisée est du BiCMOS 0,8 μ m avec 11mm² pour un circuit analogique comportant un neurone.

¹¹Very Large Scale Integration

La version actuelle, Gaillimh, calcule la plasticité grâce au FPGA qui gère également le réseau. De plus, le paramétrage des conductances constituant chaque neurone permet de simuler plusieurs types de neurones. Ajoutons que la technologie utilisée est SiGe $0,35\mu m$ avec $10,5mm^2$ pour un circuit pouvant comporter 5 neurones. Nous reviendrons plus en détail sur ces plateformes puisqu'il s'agit des supports utilisés pour réaliser les travaux de cette thèse. Ici l'accent est mis sur la précision du modèle et la vitesse de fonctionnement cadencée au temps réel biologique.

1.1.4 Conclusion

Nous avons évoqué le fait que les méthodes de calcul basées sur un paradigme issu de la machine de Turing trouvent leurs limites dans le traitement qui se veut concurrent aux capacités du vivant. Les architectures d'aujourd'hui, comme nous l'avons vu, tendent à se baser sur de nouveaux paradigmes importés de l'observation du fonctionnement des organismes biologiques et plus particulièrement des systèmes nerveux.

Nous allons dans la prochaine partie voir l'environnement dans lequel les travaux de cette thèse ont été menés en présentant notre équipe et le domaine dans lequel elle évolue pour le développement de technologies bio-inspirées simulant le vivant ou l'assistant. Nous verrons que les réalisations allant dans ce sens sont très encouragées et trouvent un espace d'expression qui ne demande qu'à être alimenté.

1.2 Environnement des travaux de thèse

1.2.1 Présentation de l'équipe "Ingénierie des Systèmes Neuromorphiques"

Affiliation

L'équipe *Ingénierie des Système Neuromorphiques* ou ISN faisait partie du groupe COFI (COception FIabilité) jusqu'au 31/12/2008 au sein du laboratoire IMS¹² (Laboratoire de l'Intégration du Matériau au Système). Depuis le 1er janvier 2009, l'équipe a rejoint le groupe Bio-électronique dans le cadre d'un regroupement thématique.

¹²Le laboratoire IMS est le fruit du regroupement de 3 laboratoires (IXL, LAPS et PIOM) en 2007. Nous sommes issus du laboratoire de microélectronique IXL.

L'équipe ISN compte aujourd'hui 5 enseignants-chercheurs, 5 doctorants, 3 ingénieurs. Concernant son activité, elle développe principalement des systèmes électroniques en s'inspirant des systèmes nerveux du vivant, et des solutions pour des applications biomédicales. L'équipe a deux principaux objectifs. D'une part, développer des circuits analogiques simulant un modèle électrique de l'activité du neurone. D'autre part, produire des outils matériels et logiciels d'instrumentation de réseaux de neurones artificiels et biologiques.

Elle travaille en collaboration avec d'autres laboratoires français et internationaux. Ces collaborations sont très fortes et trans-disciplinaires. Ainsi interviennent des acteurs allant de la neurophysiologie à l'informatique sans oublier l'électronique. Par son domaine d'activité, l'équipe est en synergie avec différents disciplines et se retrouve donc souvent à collaborer à des projets inter-disciplinaires régionaux, nationaux, européens, ainsi qu'internationaux.

Projets neuromorphiques

Dans ces projets l'équipe apporte une expertise en concevant des circuits intégrés neuromorphiques.

- Sensemaker (2002-2005)

Ce projet regroupant 5 groupes de recherche européens (France, Allemagne, Angleterre et Irlande) avait pour objectif d'une part de combiner des approches de biologie, de physique et d'ingénierie afin de réaliser un système de perception multisensorielle adaptatif. Et d'autre part, la volonté était d'améliorer les connaissances des systèmes naturels et les liens entre les mondes de la science des mathématiques et celui de la biologie.

- Facets (2005-2009)

Ce projet regroupe une douzaine d'équipes européennes de divers horizons. Cette thèse a été financée par ce projet. Nous détaillerons donc ce dernier au paragraphe 1.2.2.

Projets d'instrumentation et d'interfaçage

Ces projets regroupent les activités de l'équipe liées au développement d'une instrumentation qui vise à créer des interfaces vivant-artificiel.

- Neurobit (2002-2005)

Ce projet avait pour vocation de coupler un robot à des neurones biologiques *in-vitro* afin d'étudier les mécanisme d'adaptation et d'apprentissage. Ce projet a regroupé l'expertise des domaines de l'électrophysiologie, de la neurobiologie, de la neuroingénierie, de la microélectronique et des technologie de l'information.

- Neuro-vers-IT (2006-2009)

Ce projet fait collaborer des chercheurs de six pays européens des domaines des neurosciences, des sciences cognitives, de la robotique, de la microélectronique et de la recherche biomédicale. L'objectif ici est de contribuer à l'avancée des domaines de la robotique bio-inspirée et de la recherche biomédicale. L'équipe ISN est en charge du développement des interfaces hybrides vivant-artificiel.

- Delivrer (2007-2010)

Ce projet réunit des équipes de laboratoires d'universités bordelaises dans le but de développer une instrumentation capable de contrôler en boucle fermée une libération d'insuline chez les diabétiques. L'insuline est l'hormone responsable de la concentration de glucose dans le sang. La régulation se fait à partir de l'enregistrement et l'analyse de l'activité de cellules bêta-pancréatiques cultivées *in-vitro*.

- STN oscillations (2009-2012)

Ce projet financé par l'ANR réunit 2 équipes de laboratoires bordelais. La collaboration part du constat qu'une activité dans une certaine gamme de fréquences est observée dans un noyau subthalamique chez les patients atteints de tremblements symptomatiques de la maladie de Parkinson. L'un des objectifs est d'observer la validité de ce constat en provoquant artificiellement cette activité oscillatoire dans le cerveau de rats ou de singes sains. L'autre objectif, thérapeutique, est de tenter de rétablir une activité normale dans le noyau subthalamique incriminé chez des singes atteints de la maladie de Parkinson. L'équipe ISN a pour rôle de développer l'instrumentation nécessaire pour ces expériences et de développer un dispositif analogique (sur silicium) afin de préparer la production de systèmes implantables et autonomes.

1.2.2 Projet pluridisciplinaire FACETS et historique

Ces dernières années, le "Cerveau" a occupé une place de plus en plus prépondérante dans notre société. Ainsi, d'importants pôle se forment en

France (déjà à Bordeaux le Neurocampus regroupant 4 centres et instituts) et dans d'autres pays européens (comme par exemple le nouveau Stockholm Brain Institute en Suède). D'importants projets voient le jour et la communauté neuro-pluri-disciplinaire¹³ internationale ne cesse d'intensifier les activités de recherches touchant de près ou de loin au système nerveux.

FACETS¹⁴ est le nom d'un projet quadriennal européen financé par l'Union Européenne débuté en septembre 2005 et regroupant des scientifiques de différents horizons. Ce projet s'inscrit dans la thématique des FET¹⁵ concernant les futures technologies émergentes. Les domaines impliqués gravitent autour des sciences, technologies et traitement de l'information appliqués aux neurosciences en incluant la neurophysiologie, les neurosciences computationnelles et les systèmes neuromorphiques (systèmes électroniques dédiés à la simulation neuronale). L'objectif est de développer des outils basés sur de nouveaux paradigmes bio-inspirés et de pousser plus loin notre connaissance des propriétés du cerveau. Le projet FACETS tend à faire avancer la question centrale mais toujours en suspens concernant les mécanismes fonctionnels du cerveau. Ainsi des neuroscientifiques, des informaticiens, des modélisateurs, des ingénieurs ainsi que des physiciens se regroupent pour améliorer nos connaissances et nos techniques. Le consortium est formé de 13 équipes comptant près de 80 chercheurs européens d'Autriche, de France, d'Allemagne, de Hongrie, de Suède, de Suisse et d'Angleterre.

Les équipes se répartissent en 3 niveaux et 2 échelles. On distingue tout d'abord le niveau biologique, le niveau de modélisation et simulation logicielle et enfin le niveau de simulation électronique. Puis nous avons l'échelle cellulaire d'un côté et de l'autre l'échelle du réseau. Le premier s'intéresse à une description précise de quelques cellules nerveuses tandis que le second s'intéresse aux dynamiques de gros réseaux de neurones (à partir de quelques milliers). La figure 1.3 présente l'organigramme de cette organisation avec ses interactions. L'équipe ISN de Bordeaux se situe à l'échelle cellulaire au niveau simulation matérielle.

¹³Permettez l'emploi d'un néologisme.

¹⁴Fast Analog Computing with Emergent Transient States

¹⁵Future Emergent Technologies

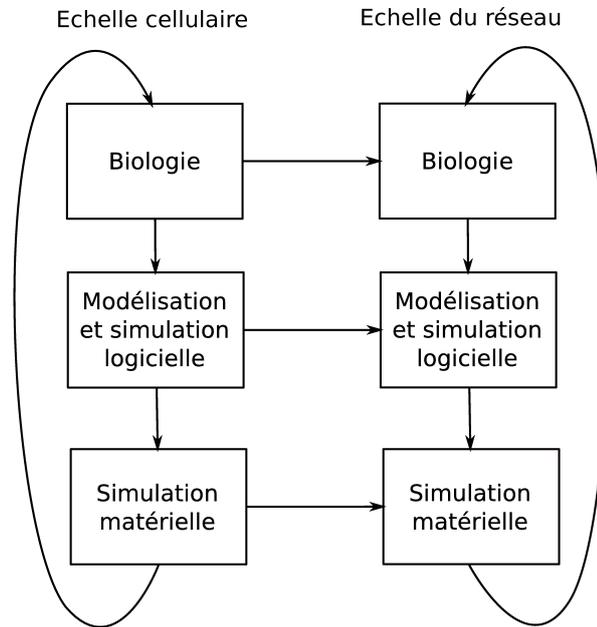


FIG. 1.3 – Structure du projet FACETS.

1.2.3 Modélisation

La modélisation des systèmes nerveux s'est très diversifiée grâce aux importants progrès aussi bien techniques que scientifiques dans la compréhension du vivant comme nous l'avons déjà évoqué. Les outils neuromorphiques développés dans l'équipe ISN fournissent un important matériel de simulation et d'exploration pour les neurosciences intégratives et computationnelles. Ces outils ont été utilisés afin de comprendre comment évolue la plasticité dans un contexte particulier. Précisons le cadre de l'étude. Des neurones proches géographiquement devraient en principe recevoir un grand nombre d'afférences communes et par conséquent une activité relativement corrélée. En effet les données anatomiques indiquent que certaines régions corticales présentent une structuration spatiale. Citons par exemple l'organisation rétinotopique ou tonotopique dont les neurones possèdent des propriétés locales identiques. Les délais de transmission du message nerveux entre neurones proches sont en-dessous de la milliseconde, la vitesse de propagation étant en moyenne de l'ordre de 10 m.s^{-1} [39]. Les neurones communiquent à travers la propagation des PAs au sein d'un réseau et leur communication est

régulée par la plasticité synaptique. Les neurones proches géographiquement sont supposés recevoir des informations corrélées et devraient donc avoir une activité corrélées. Nous nous sommes intéressés à étudier le mode d'évolution des poids synaptiques au sein d'un petit réseau de neurones, simulés par un système neuromorphique. Ces neurones sont stimulés, outre leurs connexions au sein du réseau, par des informations extérieures dont la corrélation est modulée.

1.2.4 conclusion

Nous allons maintenant voir plus en détail, le modèle du neurone et de la synapse qui ont servi de base à l'implantation matérielle utilisée, ainsi que le contexte expérimental qui a donné lieu aux études présentées dans cette thèse.

1.3 Modèles

1.3.1 Physiologie du Neurone

Anatomie, membrane

Un neurone (figure 1.4) est composé d'un corps cellulaire encore appelé *soma*, ainsi que de 2 catégories de neurites : l'*axone* et les *dendrites*. Les dendrites, afférentes, lui permettent de recevoir des informations provenant d'autres cellules (neurones, cellules sensorielles). L'axone, efférent, envoie des informations vers d'autres cellules (neurones, muscles). Les neurones peuvent traiter et convoier de l'information essentiellement sous forme de messages nerveux. L'état de sortie du neurone dépend de son état interne lui-même dépendant de ses entrées. Le long de l'axone se propage le message nerveux, encore appelé Potentiel d'Action¹⁶ qui est un événement stéréotypé (figure 1.5). La transmission de l'information s'effectue au niveau de la synapse (encadré de la figure 1.4), où le PA libère des neurotransmetteurs qui se logent sur les récepteurs du neurone postsynaptique, ce qui modifiera le potentiel postsynaptique.

L'idée générale est que le potentiel de membrane du soma varie en fonction des potentiels postsynaptiques intégrés au niveau de ses synapses afférentes dendritiques ou somatiques. Ces potentiels postsynaptiques peuvent

¹⁶ ce sont les « *spikes* » des modèles neuro-mimétiques du §1.1.3

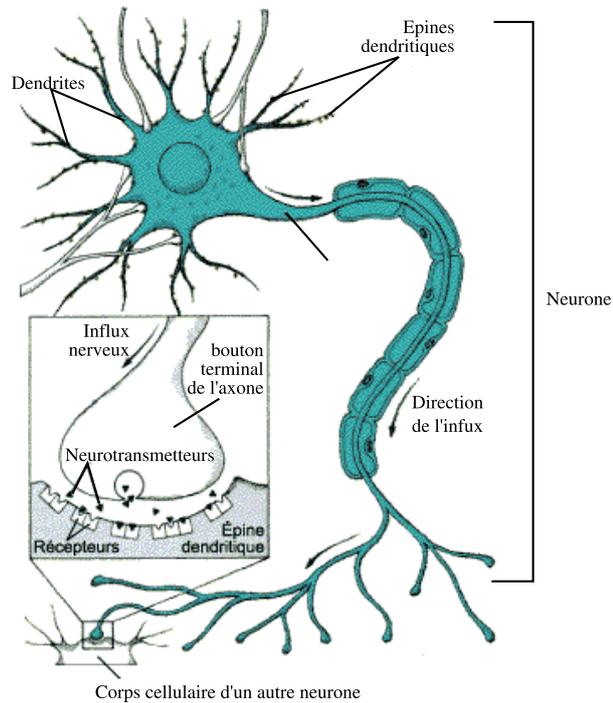


FIG. 1.4 – Schéma d'un neurone avec le corps cellulaire ou soma, les dendrites et l'axone avec ses ramifications. *Encadré* : une synapse est représentée avec la partie présynaptique constituée du bouton terminal de l'axone du neurone présynaptique. La partie postsynaptique est représentée par une épine dendritique comportant des récepteurs sur la membrane postsynaptique. Entre les deux, se situe la fente synaptique par laquelle transitent les neurotransmetteurs.

être excitateurs ou inhibiteurs et vont respectivement augmenter ou diminuer le potentiel global. Lorsque le potentiel de membrane atteint un certain seuil, il se décharge, et un PA se propage le long de l'axone puis le long de ses ramifications jusqu'à atteindre les synapses.

Membrane cellulaire et potentiel d'action

La membrane du neurone est organisée en bicouche de phospholipides qui isole le milieu intracellulaire du milieu extracellulaire. La membrane contient cependant des protéines transmembranaires ayant un accès à l'intérieur ainsi qu'à l'extérieur (figure 1.6). Il existe principalement trois catégories de protéines transmembranaires : les pompes de sodium-potassium, les canaux potassiques ainsi que les canaux sodiques. Le milieu intérieur de la cellule,

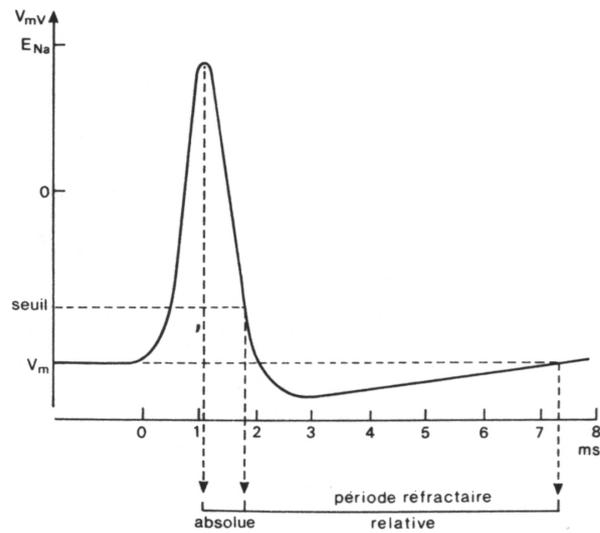


FIG. 1.5 – Message nerveux, communément appelé *Potentiel d'Action* (PA).

appelé cytosol, ainsi que le milieu extérieur contiennent des ions en solution dans de l'eau (composant principal des deux milieux). Les pompes Na^+/K^+ fonctionnent en permanence, utilisant de l'énergie, pour transporter les ions Na^+ du cytosol vers l'extérieur et inversement les ions K^+ de l'extérieur vers l'intérieur. Ce mécanisme permet de maintenir les gradients de concentration

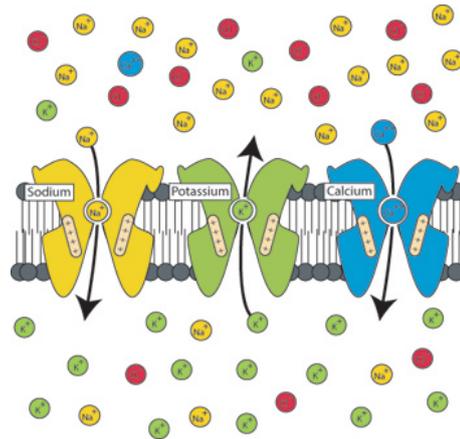


FIG. 1.6 – Membrane du neurone composée d'une bicouche de phospholipides. Des canaux transmembranaires permettent le passage sélectif de certains ions.

responsables du potentiel de repos de la membrane (environ -65 mV). Ainsi, le milieu intérieur est beaucoup plus riche que le milieu extérieur en ions K^+ , et inversement, l'extérieur est plus riche en ions Na^+ . Les canaux ioniques potassique et sodique, sont des canaux transmembranaire qui peuvent être dans différents états : soit ouverts soit fermés, soit, en plus pour le sodium, inactivés. L'état de ces canaux ioniques est dépendant du potentiel de membrane. On les nomme pour cela des canaux voltage-dépendant. En effet, lorsque le potentiel de membrane atteint un certain seuil de déclenchement (autour de -40 mV), suffisamment de canaux sodiques sont alors ouverts et les ions Na^+ pénètrent dans le cytosol. Ceci entraîne une rapide dépolarisation de la membrane qui amène le potentiel de membrane au-dessus de 0 mV. Puis les canaux sodiques passent à l'état inactivé. Les canaux potassiques s'ouvrent alors et les ions K^+ sortent de la cellule, permettant au potentiel de membrane de redevenir négatif. A ce moment, la perméabilité de la membrane au sodium est très faible alors qu'elle est très forte pour le potassium ce qui amène le potentiel à tendre vers le potentiel d'équilibre du potassium après une phase d'hyperpolarisation.

Modèle de Hodgkin et Huxley

Nous avons vu dans la première partie de ce chapitre que les modèles de neurones appliqués au calcul étaient très simples puis se sont complexifiés. Mais dès l'apparition des premiers modèles, certains s'étaient attelés à définir un modèle biophysique du neurone. Ainsi, les physiologistes Hodgkin et Huxley, dans une série de publications à partir de 1949, regroupées et synthétisées en 1952 [27] exposent un modèle du comportement électrique de la membrane du neurone réalisé grâce à la technique de potentiel imposé (plus connu sous le terme de *voltage-clamp* en anglais) appliquée à l'étude de l'axone géant du calmar. Ce modèle permet de reproduire le potentiel d'action grâce à la contribution du potassium, du sodium et d'autres ions. Le potentiel de membrane, défini par la différence de potentiel intra- et extracellulaire, est dû à la différence de concentration ionique entre l'intérieur et l'extérieur de la cellule. L'évolution de ce potentiel est due aux mouvements ioniques à travers des protéines transmembranaires qu'on appelle également canaux ioniques. Dans leur modèle, le modèle électrique de la membrane est représenté par le circuit de la figure 1.7. Le courant est transporté à travers la membrane par la charge d'une capacité et par les mouvements ioniques

à travers des résistances mises en parallèle du condensateur. Le courant ionique est divisé en courants sodique et potassique (I_{Na} et I_K), ainsi qu'en un courant de fuite (I_{Fuite}). Le courant total de membrane est alors la somme

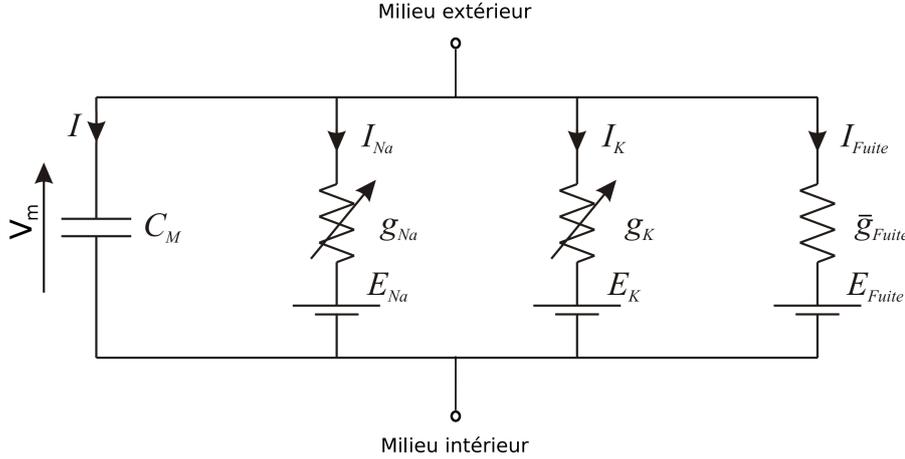


FIG. 1.7 – Schéma électrique du modèle de Hodgkin et Huxley avec des conductances sodique I_{Na} , potassique I_K et de fuite I_{Fuite} .

d'un courant de capacité et d'un courant ionique comme dans l'équation suivante :

$$C_M \frac{dV_m}{dt} = -I_i \quad (1.1)$$

où C_M est la capacité de membrane par unité de surface (supposée constante), $\frac{dV_m}{dt}$ est la variation de potentiel de membrane V_m par rapport à sa valeur de repos, t est le temps et enfin où le courant ionique I_i est équivalent à :

$$I_i = I_{Na} + I_K + I_{Fuite} \quad (1.2)$$

La perméabilité ionique de la membrane est représentée par les conductances ioniques g_{Na} , g_K et \bar{g}_{Fuite} . De plus les potentiels d'équilibres des trois espèces ioniques sont E_{Na} , E_K et E_{Fuite} pour respectivement le potassium, le sodium et enfin le dernier pour la fuite. Ainsi, les différents courants ioniques sont :

$$I_{Na} = g_{Na}(V_m - E_{Na}), \quad (1.3)$$

$$I_K = g_K(V_m - E_K), \quad (1.4)$$

$$I_{Fuite} = \bar{g}_{Fuite}(V_m - E_{Fuite}) \quad (1.5)$$

Dans les équations précédentes, E_{Na} , E_K , E_{Fuite} , C_M et \bar{g}_{Fuite} sont constantes, alors que g_{Na} et g_K sont fonction du temps et du potentiel de membrane V_m . Les expressions des deux dernières conductances sont fournies dans les équations 1.6 et 1.7 respectivement.

$$g_{Na}(V_m) = \bar{g}_{Na} \cdot m^3(V_m) \cdot h(V_m) \quad (1.6)$$

$$g_K(V_m) = \bar{g}_K \cdot n^4(V_m) \quad (1.7)$$

où les \bar{g}_x représentent les conductances maximales, m^3 et n^4 sont des variables d'état représentant les proportions de canaux activés et h de canaux inactivés. Un neurone possédant les trois types de canaux précédents, présente un comportement correspondant à celui d'un neurone typique inhibiteur du cortex que l'on nomme neurone FS¹⁷. Afin de modéliser le comportement d'un neurone excitateur, nommé neurone RS¹⁸ [40], il convient d'ajouter un canal dit modulant dont le courant est fournit dans l'équation 1.8 où \bar{g}_{mod} est la conductance maximal du canal et la variable d'état m la proportion de canaux à l'état actif.

$$I_{mod} = \bar{g}_{mod} \cdot m(V_m)(V_m - E_{mod}) \quad (1.8)$$

Les variables d'état sus-citées, dépendant du potentiel V_m , répondent à l'équation différentielle suivante :

$$\tau \cdot \frac{dx}{dt}(t) = x_\infty(V_m) - x(t) \quad (1.9)$$

avec la fonction sigmoïde

$$x_\infty(V_m) = \frac{1}{1 + \exp(\pm \frac{V_m - V_{offset}}{V_{pente}})} \quad (1.10)$$

où x est la variable d'état, le signe retenu à l'intérieur de l'exponentielle sera - pour l'activation et + pour l'inactivation. Les paramètres V_{offset} et V_{pente} sont respectivement le décalage et la pente de la sigmoïde.

¹⁷Fast Spiking

¹⁸Regular Spiking

1.3.2 Synapses

Anatomie synaptique, physiologie

La synapse, comme nous l'avons évoqué, est le lieu de transmission du message nerveux entre deux neurones (figure 1.4).

Il existe deux principales catégories de synapses : les synapses électriques (appelées en anglais *gap junctions*) et les synapses chimiques (les plus répandues dans le cerveau). Nous nous intéresserons uniquement aux synapses chimiques étant donné que ce sont celles qui manifestent des mécanismes de plasticité. Au niveau de la synapse, on compte trois parties. Tout d'abord, nous avons le bouton présynaptique constitué par la terminaison de l'axone du neurone présynaptique. Puis nous trouvons la fente synaptique qui est l'espace entre les deux neurones connectés. Et enfin, la troisième partie est la membrane postsynaptique, comportant des récepteurs de la dendrite ou du soma du neurone postsynaptique.

Ainsi, lorsque le PA du neurone présynaptique arrive au niveau du bouton présynaptique, des neurotransmetteurs (excitateur/inhibiteur) sont libérés dans la fente synaptique et se fixent sur les récepteurs de la membrane postsynaptique. Ceci a pour effet d'initier un potentiel postsynaptique qui se propage jusqu'au soma pour contribuer au potentiel global de la membrane du neurone postsynaptique.

Le bouton présynaptique, possède dans le cytosol des vésicules contenant les neurotransmetteurs. La dépolarisation de la membrane par un PA a pour effet d'ouvrir des canaux calciques transmembranaires dépendant du potentiel (à l'instar des canaux sodiques) entraînant l'entrée d'ions Ca^{2+} entraînant la fusion des vésicules avec la membrane présynaptique. Cette fusion a pour effet de libérer les neurotransmetteurs dans la fente synaptique, processus connu sous le nom d'*exocytose*. Il existe trois principales catégories chimiques de neurotransmetteurs et neuromodulateurs : les acides aminés, les amines et les peptides. Nous nous intéresserons uniquement à la transmission de l'information et donc aux neurotransmetteurs qui sont l'acide γ -aminobutyrique (GABA) et à l'acide glutamique encore appelé glutamate (Glu), respectivement l'inhibiteur et l'excitateur du cerveau.

La fente synaptique a une largeur de 20 à 50 nm. Cet espace est rempli par des protéines fibreuses permettant l'adhésion des membranes pré- et postsynaptiques. C'est dans cet espace que les neurotransmetteurs vont aller

de la membrane présynaptique aux récepteurs postsynaptiques.

La membrane postsynaptique possède des récepteurs qui s'activent lorsque des neurotransmetteurs se fixent dessus. Il existe une multitude de récepteurs dépendants du type de neurotransmetteurs. On distingue cependant les deux grandes catégories suivantes : les récepteurs-canaux et les récepteurs couplés aux protéines G. Les premiers sont dits *ionotropiques* et les seconds *métabotropiques*.

Les récepteurs ionotropiques sont des protéines transmembranaires qui activées vont permettre le passage de certains ions. Le glutamate se fixe sur des récepteurs-canaux permettant le passage de sodium ce qui implique une dépolarisation de la membrane postsynaptique et par conséquent une excitation. Cette dépolarisation est connue sous le nom de potentiel postsynaptique excitateur (PPSE). Nous connaissons pour le glutamate surtout deux types de récepteurs-canaux : les récepteurs AMPA et NMDA. Les récepteurs AMPA sont perméables à Na^+ et K^+ , les NMDA sont en plus perméables à Ca^{2+} et dépendant du potentiel de membrane. Alors que les récepteurs AMPA sont responsables de la dépolarisation, les récepteurs NMDA en laissant entrer le calcium peuvent impliquer plusieurs fonctions neuronales et cellulaires. Le calcium pourrait participer aux mécanismes de plasticité et de mémoire. Les NMDA sont activés seulement s'il y a une dépolarisation suffisante et la présence de glutamate évidemment.

Le GABA se fixe sur des récepteurs permettant le passage de chlore, ce qui entraîne une hyperpolarisation de la membrane postsynaptique et donc une inhibition. Cette hyperpolarisation est connue sous le nom de potentiel postsynaptique inhibiteur (PPSI).

Les récepteurs métabotropiques ont un effet différent des ionotropiques. En effet, ils ont un effet plus lent et ne participent pas à la transmission nerveuse mais plutôt à des changements métaboliques en déclenchant une cascade de réactions.

Modèles synaptiques

Dans la plupart des modèles neuronaux, les événements synaptiques étaient représentés par une conductance synaptique variant en fonction du temps de façon stéréotypée suivant en général une fonction alpha [41] qui représente les PPSE. L'approximation de la conductance synaptique par ce type de fonction a été établi à partir d'enregistrements physiologiques et

non d'un modèle des mécanismes biophysiques. Les courbes de conductance sont sommées pour représenter plusieurs événements synaptiques. Dans ces modèles, aucun phénomène de saturation de la conductance n'est pris en compte. Une autre approche, plus réaliste, consiste à utiliser un modèle cinétique. Ceci fournit en plus des mécanismes biophysiques, une cohérence avec le formalisme de description des conductances de canaux ioniques. Ainsi, Destexhe et al. 1998, proposent un modèle analytique de la transmission synaptique prenant en compte la cinétique liée aux récepteurs postsynaptiques, et une simplification de ce modèle adaptée à la simulation [42, 43]. Les types de récepteurs AMPA/NMDA et $GABA_A$ présentent des potentiels postsynaptiques (PPS) ayant des amplitudes proportionnelles au nombre de PAS présynaptiques. De plus, les PPS peuvent être sommés ce qui est particulièrement pratique pour la simulation de réseaux de neurones aux dimensions variables. Ce modèle est par conséquent appelé modèle de multisynapse [44, 43] et permet de simuler l'intégration de plusieurs événements synaptiques au sein d'une seule synapse, et plus particulièrement de la membrane postsynaptique. En d'autres termes, le neurone est modélisé avec une seule synapse en entrée, à laquelle plusieurs autres neurones peuvent être connectés. Ceci peut être pratique pour la simulation électronique de synapse avec la possibilité de connecter autant de neurones présynaptiques que l'on souhaite à une unique entrée postsynaptique. Afin de simuler ce genre de modèle avec des circuits électroniques, une adaptation consiste à représenter l'exponentielle décroissante par une fonction qui décrit la conductance synaptique comme suit :

$$g = A. \exp \frac{-(t - t_0)}{\tau} \quad (1.11)$$

Une des propriétés de l'exponentielle est alors utilisée afin de sommer plusieurs exponentielles décroissantes ayant la même constante de temps. Quelque soient les réels A , B , t_a et t_b , il existe deux réels C et t_c tel que :

$$A. \exp \frac{-(t - t_a)}{\tau} + B. \exp \frac{-(t - t_b)}{\tau} = C. \exp \frac{-(t - t_c)}{\tau} \quad (1.12)$$

Ainsi il est possible de représenter un ensemble de synapses d'un même type (excitatrice/inhibitrice, ici termes en A et B) avec un seul canal (ici terme en C). Les événements synaptiques de même nature étant alors ajoutés au canal post-synaptique. Ceci constitue la multisynapse. L'inconvénient de cette méthode est de ne pas connaître la provenance d'un événement synaptique,

inconvéniént pallié par le stockage de cette information à l'extérieur du neurone. Au final nous pouvons représenter l'intégration du courant synaptique par une équation suivant le même formalisme que celui des canaux ioniques (équation 1.13).

$$\begin{cases} I_{syn} &= g_{syn}(t) \cdot (V_m - E_{syn}) \\ g_{syn}(t) &= \bar{g}_{syn} \cdot r(t) \end{cases} \quad (1.13)$$

Dans cette équation, \bar{g}_{syn} est la conductance maximale du canal synaptique. La fonction $r(t)$ suit justement une fonction décroissante du type de l'équation 1.11. Et dans le cas d'une multisynapse, l'expression de $r(t)$ est fournie dans l'équation 1.14 pour n événements se produisant aux temps $t_i \in \{1..n\}$, où $\omega_{syn}(t)$ représente le poids synaptique qui peut varier dans le temps, et où τ_{syn} est la constante de temps de la décroissance du courant synaptique.

$$\begin{cases} g_{syn}(t) &= \bar{g}_{syn} \cdot \sum_{i=1}^n r_i(t) \\ r_i(t) &= 0 & \forall t < t_i \\ r_i(t) &= \omega_{syn}(t_i) \cdot \exp\left(-\frac{t-t_i}{\tau_{syn}}\right) & \forall t \geq t_i \end{cases} \quad (1.14)$$

Etant dans un modèle biophysique, la croissance du courant n'est pas décrite comme instantanée mais comme ayant une vitesse constante. Le temps de montée¹⁹ correspondant va donc déterminer la valeur atteinte (limitée par \bar{g}_{syn}). C'est finalement ce temps de montée qui représentera le poids ou la force du courant synaptique²⁰ qui sera injecté (figure 1.8). Ainsi, l'information présynaptique sera numérisée et présentée sous la forme d'un créneau de tension de durée fixe. Le courant synaptique est directement ajouté aux autres courants ioniques de l'équation 1.2, comme suit :

$$I_i = I_{Na} + I_K + I_{Fuite} + I_{syn} \quad (1.15)$$

Notons qu'il faut ajouter à cette équation le courant du canal modulant dans le cas d'un neurone RS.

¹⁹Le temps de montée sera défini ici comme la durée durant laquelle I va de 0 à I_{max}

²⁰ou encore l'efficacité synaptique

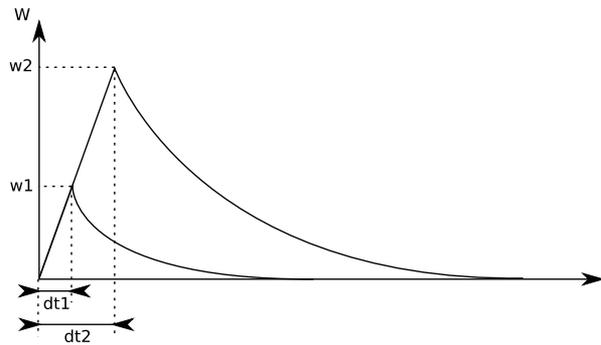


FIG. 1.8 – Les valeurs du poids synaptique dépendent du temps de stimulation de la synapse. Dans cet exemple, lorsque le temps de montée est $dt1$, le poids synaptique correspondant est alors $w2$, et de la même manière il sera de $w2$ pour un temps de montée $dt2$. Ainsi le temps de montée représente le poids synaptique.

1.3.3 Plasticité (STDP)

Comme nous l'avons évoqué précédemment, la STDP est un modèle de plasticité synaptique se basant sur la différence des temps d'émission de PA par les deux neurones concernés [45, 46, 47, 48, 49]. Cette règle dérive des études liées aux phénomènes de LTP et de LTD [50].

L'idée générale est qu'en se référant à une connexion synaptique entre deux neurones, si le neurone présynaptique émet un PA avant le neurone postsynaptique dans une fenêtre temporelle suffisamment petite, l'efficacité synaptique est alors augmentée (mécanisme de LTP). Alors que si le neurone postsynaptique émet un PA avant le présynaptique, dans une courte succession temporelle, l'efficacité de la synapse est dans ce cas réduite (mécanisme de LTD).

Ce modèle de plasticité résulte d'expérimentations menées à l'échelle de PAs individuels et a été observé dans différentes structures comme le cortex [51], l'hippocampe [52, 53] et dans des cellules en cultures [54].

Ce modèle a été enrichi dans des travaux plus récents. Citons l'introduction dans le modèle de l'effet des PA précédents d'un même neurone [55], ou encore l'effet de la localisation de la synapse sur l'efficacité synaptique [56, 57]. D'autres modèles de STDP proposent d'utiliser des règles différentes en fonction de la valeur de l'efficacité synaptique [54, 58]. D'un point de vue fonctionnel, la STDP est connue pour augmenter l'efficacité synaptique de

neurones synchronisés [59, 50]. Ajoutons qu'elle semble jouer un rôle dans la synchronisation des assemblées de neurones [2]. En fonction de la forme de la STDP, le comportement du réseau de neurones soumis peut varier en fonction du fait que le ratio entre LTP et LTD peut influencer la convergence des valeurs d'efficacité synaptique [50].

Mécanismes

Comme l'illustre le schéma A de la figure 1.9, la synapse reliant le neurone présynaptique j au neurone postsynaptique i , possède une certaine efficacité, dite poids de la synapse W_{ji} et représentée par une conductance synaptique. Dans le schéma B de la figure, on peut voir deux cas d'activité des neurones j et i . Dans le cas B1, le neurone j émet un PA au temps t_j avant que le neurone i n'émette un PA au temps t_i . La différence temporelle de ces événements est $Dt_1 = t_i - t_j$ et $Dt_1 > 0$. Ce cas entraîne une potentialisation²¹ et le poids W_{ji} devient $W_{ji} = W_{ji} + y_1$ (y_1 de C de la figure). Dans le cas B2, $Dt_2 = t_i - t_j$ et $Dt_2 < 0$ ce qui entraîne une dépression²² avec $W_{ji} = W_{ji} + y_2$ (y_2 de C de la figure).

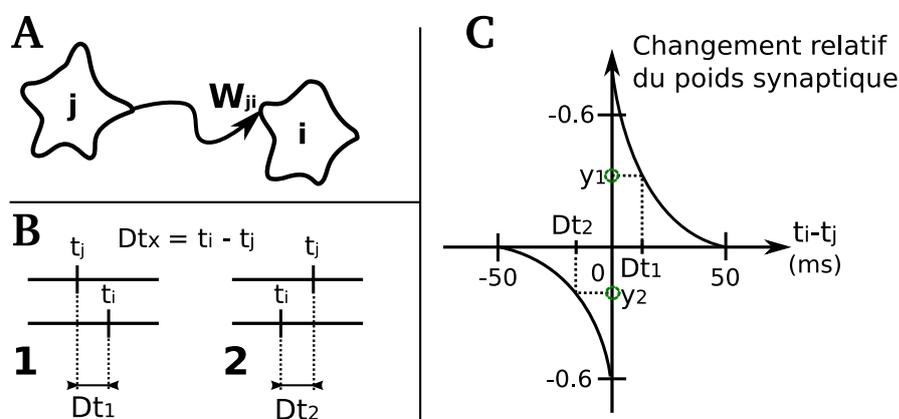


FIG. 1.9 – Illustration du mécanisme de la STDP. Un neurone présynaptique j est connecté à un neurone postsynaptique i . W_{ji} est l'efficacité de la synapse entre j et i . Si l'émission de PA par les deux neurones se fait dans l'ordre j puis i , alors W_{ji} est augmenté de la quantité $|y_1|$, dans le cas de l'ordre inversé, W_{ji} est diminué de $|y_2|$.

²¹potentialisation = augmentation du poids synaptique

²²dépression = diminution du poids synaptique

Modèle de STDP utilisé

Le modèle de plasticité que nous utilisons dans nos travaux est un modèle de STDP basé sur le modèle biophysique présenté par Badoual et al. 2006 [60]. Le modèle a été choisi dans le cadre de FACETS car il rend compte de mécanismes biophysiques précis. Ce modèle biophysique part de la modélisation d'un neurone composé d'une dendrite, d'une épine dendritique, d'un soma et d'un axone. Le modèle de canaux ioniques suit le formalisme de Hodgkin et Huxley. Des canaux potassiques dépendant-Calcium et des canaux calciques haut-voltage activés ont été ajoutés à la dendrite. Les processus liés aux échanges de Calcium à travers des canaux voltage-dépendants et des pompes ont été représentés. La cinétique des récepteurs dendritiques AMPA et NMDA a été incluse dans le modèle. Le compartiment postsynaptique des épines possède également des mécanismes de régulation de l'efficacité synaptique en utilisant des enzymes de LTP et de LTD. La plasticité totale est donc évaluée comme étant la différence entre la contribution de la LTP et celle de la LTD.

Ainsi, le modèle biophysique prend en compte l'influence du calcium ainsi que d'enzymes supposées être les initiatrices de la LTP et de la LTD. Ce modèle complexe étant satisfaisant d'un point de vue physiologique nécessite néanmoins plusieurs équations différentielles pour chaque synapse. Son utilisation devient donc très limitée dans le cadre de la simulation d'un réseau de neurones. Par conséquent les auteurs en ont proposé un modèle simplifié se basant sur les temps d'émission de PAs.

Des fonctions exponentielles remplacent les exponentielles décroissantes de déclenchement des enzymes responsables de la LTP et de la LTD. Ainsi la quantité et le signe de la plasticité sont exprimés par deux fonctions exponentielles P et Q déclenchées respectivement par les PAs pré- et postsynaptiques. Ceci permet d'établir une correspondance entre l'évolution de l'efficacité synaptique et les temps d'émission des PAs pré- et postsynaptiques. Cette correspondance peut être écrite comme suit :

$$\frac{d\omega_{ji}}{dt} = \sum_k P[t - \tilde{t}_j(t)]\delta(t - t_{i,k}) - \sum_l Q[t - \tilde{t}_i(t)]\delta(t - t_{j,l}), \quad (1.16)$$

où ω_{ji} est l'efficacité synaptique (également nommé poids synaptique) de la synapse connectant le neurone présynaptique j au neurone postsynaptique

i . La somme s'opère sur l'ensemble des temps d'émission de PAs postsynaptiques $t_{i,k}$ et présynaptiques $t_{j,l}$ où $\tilde{t}_i(t)$ indique le temps du dernier PA du neurone i au temps t . $k \in \{1..k\}$ et $l \in \{1..l\}$ sont les rangs dans l'historique des PAs des neurones respectifs i et j . Les fonctions P et Q décrivent respectivement la quantité de LTP et de LTD en fonction de la différence temporelle entre les PAs des neurones pré- et postsynaptiques, et s'énoncent ainsi :

$$P(t) = \exp[-t/\tau_P], \quad Q(t) = \exp[-t/\tau_Q] \quad (1.17)$$

où τ_P et τ_Q sont les constantes de temps de la LTP et de la LTD.

Le modèle intègre deux mécanismes supplémentaires. Le premier prend en compte ce que l'on appelle les interactions de triplets. En effet, des études ont montré que l'interaction non-linéaire entre des paires successives de PAs pré- et postsynaptiques, peut être représentée par un facteur dit de suppression. Ce facteur de suppression permet de moduler la quantité de variation de l'efficacité synaptique en fonction de l'historique des PAs précédents et ce au niveau de chaque PA. Il est exprimé comme suit :

$$\epsilon_i = 1 - \exp[-(t_i - t_{i-1})/\tau_s] \quad (1.18)$$

où t_i et t_{i-1} sont deux temps d'émission de PAs successifs d'un même neurone i et τ_s est la constante de temps de suppression.

Le deuxième mécanisme ajouté est la prise en compte de phénomènes de saturation. Cela afin de pallier la possibilité d'évolution de l'efficacité synaptique vers des valeurs très élevées ou à l'inverse très petites et même négatives. Ce mécanisme de saturation est assuré par l'encadrement du poids synaptique entre deux bornes inférieure ω_{LTD} et supérieure ω_{LTP} .

Le modèle simplifié est intéressant car il ne nécessite de calculs que lorsqu'un PA est émis. En regroupant les équations 1.16 à 1.18, l'équation finale simplifiée du modèle biophysique est :

$$\begin{aligned} \frac{d\omega_{ji}}{dt} = & -\epsilon_j \epsilon_i \left[(\omega_{ji} - \omega_{LTP}) \sum_k P[t - \tilde{t}_j(t)] \delta(t - t_{i,k}) \right. \\ & \left. + (\omega_{ji} - \omega_{LTD}) \sum_l Q[t - \tilde{t}_i(t)] \delta(t - t_{j,l}) \right] \end{aligned} \quad (1.19)$$

où ω_{ji} est le poids synaptique entre les neurones pré- et postsynaptiques j et i . $t_{i,k}$ et $t_{j,l}$ sont les ensembles des temps de décharge des neurones

pré- et postsynaptiques. P et Q sont les quantités de variation de LTP et de LTD avec $P(t) = A_+ \exp(-t/\tau_P)$ et $Q(t) = A_- \exp(-t/\tau_Q)$, et où $A_{+/-}$ sont des facteurs multiplicateurs ($A_{+/-} \in]0, 1[$). Les fonctions ϵ_k prennent en compte l'historique des décharges du neurone k , et sont définies par : $\epsilon_j = 1 - \exp[-(t - \tilde{t}_j(t))/\tau_{\epsilon_j}]$ et $\epsilon_i = 1 - \exp[-(t - \tilde{t}_i(t))/\tau_{\epsilon_i}]$. ω_{LTP} et ω_{LTD} sont les bornes maximale et minimale des valeurs de poids comme mentionné précédemment avec $\omega_{LTD} < \omega_{ji} < \omega_{LTP}$. Les valeurs des paramètres utilisés ci-dessus sont fournies dans le tableau 1.1.

paramètre	A_-	A_+	τ_P	τ_Q	τ_{ϵ_j}	τ_{ϵ_i}
valeur	0,1	0,05	14,8 ms	33,8 ms	28 ms	88 ms

TAB. 1.1 – Paramètres biologiques utilisés pour la STDP d'après [55]

1.3.4 Bruit

Il existe une grande variété de neurones. Dans le cortex, les neurones excitateurs sont en général des cellules pyramidales et possèdent entre 5 000 et 60 000 synapses [61, 62]. Gerstner et Kistler rapportent qu'un neurone cortical typique de vertébré connecte environ 10 000 neurones postsynaptiques [23]. Le corps cellulaire intègre donc un très grand nombre d'informations en provenance d'autres cellules nerveuses ou de cellules sensorielles. Toutefois, 70% de ces connexions proviennent de neurones corticaux [63, 64]. Des enregistrements montrent que chez l'animal éveillé, l'activité des neurones a une fréquence de décharge comprise en général entre 5 et 20 Hz [65, 66]. Un neurone est en permanence bombardé de PPSs, et ce bombardement incessant lui confère une activité propre, qui peut sembler aléatoire. Comme nous l'avons déjà mentionné, l'enregistrement de plusieurs cellules montre que des neurones peuvent pendant un temps avoir une activité corrélée malgré le bruit dû au grand nombre d'afférences. Le bruit synaptique semble influencer la variabilité des intervalles entre deux potentiels d'action (en anglais ISI²³) [67, 68].

Abeles [69] montre que 29 PAs synchrones suffisent pour faire décharger un neurone cible alors qu'il en faudrait 330 asynchrones. Il montre également que les neurones corticaux ont une activité spontanée se répartissant selon

²³Inter-Spike Interval

une loi de Poisson. De plus, à partir d'enregistrements de 6 neurones avec 1 électrode, il semblerait que tous les neurones ne puissent pas décharger simultanément, ce qui serait sinon assimilable à un fonctionnement épileptique.

Dans les travaux réalisés dans cette thèse nous utiliserons la notion de bruit synaptique pour faire référence à l'influence des stimulations de la multitude d'afférences d'un neurone. Ainsi, notre définition de bruit est particulière et sera modélisée pour un neurone par une séquences de stimulations appliquées en entrée de ce dernier.

1.3.5 Conclusion

Nous verrons dans les chapitres qui suivent une dimension fonctionnelle de neurones en réseau modulés par une règle de plasticité, et basés sur un modèle biophysiquement réaliste et précis.

1.4 Conclusion

Comme nous l'avons vu, la recherche en neurosciences intégratives et computationnelles connaît une activité intense et très diversifiée. Les travaux de cette thèse s'inscrivent dans une approche pluri-disciplinaire et profitent des avancées techniques afin de faire avancer les connaissances sur le fonctionnement du système nerveux et de préparer la place à de futurs développements technologiques inspirés du vivant.

Ainsi mes travaux, à la frontière de plusieurs disciplines, ont inclus le réglage et la configuration de circuits électroniques dédiés à la simulation d'un modèle de neurones. J'ai également programmé des outils permettant d'une part d'exploiter ces circuits, et d'autre part de les utiliser pour l'exploration de propriétés fonctionnelles de petits réseaux de neurones. Il a été question d'étudier l'influence du bruit sur la plasticité synaptique, et plus particulièrement de l'effet de la corrélation du bruit appliqué à de petits réseaux de neurones.

J'ai également développé une structure permettant l'utilisation de la plateforme de simulation via le web pour étendre l'utilisabilité de ces outils à la communauté scientifique.

Nous allons donc voir dans quelle mesure un système électronique neuro-morphique constitue un outil utile dans l'exploration de propriétés en neu-

rosiences. Le chapitre 2 présente les travaux réalisés sur un petit réseau de 6 neurones excitateurs qui nous a permis de montrer quelle convergence implique la STDP lorsque les neurones sont stimulés par des entrées corrélées, soit toutes de la même manière, soit en distinguant 2 groupes de neurones. Dans le chapitre 3 nous verrons comment régler un système neuromorphique complexe et comment l'utiliser, ainsi que sa mise à disposition de la communauté scientifique via Internet et le langage PyNN. Enfin, nous parlerons de la dissémination des travaux réalisés au sein de l'équipe et nous discuterons de l'ensemble du travail réalisé.

Chapitre 2

Bruit corrélé et plasticité

Dans ce chapitre, nous étudierons l'influence de la corrélation du bruit synaptique sur la plasticité au sein d'un réseau de neurones excitateurs.

Nous nous focaliserons sur les petits réseaux où chaque neurone est stimulé par un pattern « bruit » d'entrée en substitution à la multitude de ses connexions afférentes. Ce bruit est donc une somme de stimuli qui confère au neurone une activité qui lui est propre. Les patterns de bruit des différents neurones du réseau peuvent être corrélés entre eux. Cette corrélation fait référence au fait que des neurones peuvent partager un grand nombre d'entrées et peuvent avoir une activité synchronisée dans le cas du traitement d'informations liées, comme nous l'avons vu au chapitre précédent du manuscrit.

Ces neurones sont basés sur un modèle électrique, comme nous l'avons vu dans le chapitre 1 et simulés par des circuits analogiques. Nous analyserons plus précisément l'évolution des poids synaptiques des neurones de ce réseau lorsqu'ils sont stimulés. Les stimulations sont corrélées de façon variable et maîtrisable. De plus les synapses des neurones sont soumises à une règle de plasticité basée sur la coïncidence temporelle des potentiels d'action pré- et postsynaptiques.

Dans un premier temps nous détaillerons les caractéristiques du système de simulation utilisé. Dans un deuxième temps nous présenterons l'étude réalisée dans le cas où tous les neurones sont stimulés par des patterns de bruit ayant la même corrélation. L'analyse des résultats obtenus montrera la distribution des valeurs de poids synaptiques en fonction de l'évolution de la corrélation des patterns de bruit, ainsi que l'évolution de la corrélation entre

les trains de PAs des neurones.

Enfin, nous présenterons l'étude réalisée lorsque deux groupes de neurones du réseau sont stimulés par des patterns de bruit corrélés distinctement. Nous analyserons cette fois-ci le type de convergence des valeurs de poids synaptique pour différentes catégories de synapses (suivant leur groupe d'appartenance) en fonction de l'évolution des deux corrélations des patterns de bruit d'entrée.

2.1 Système de simulation électronique PAX

2.1.1 Propriétés du système

Le système utilisé pour ces études, nommé PAX, est composé de circuits intégrés analogiques spécifiques (ASIC¹) implantés sur une carte PCI (figure 2.1). Les ASICs simulent l'activité électrique des neurones, tandis qu'un circuit numérique programmable FPGA² gère la connectivité permettant de mettre les neurones en réseau. La plasticité synaptique n'est pas gérée par le *hardware* mais par l'ordinateur hôte en commandant le circuit FPGA (figure 2.2).

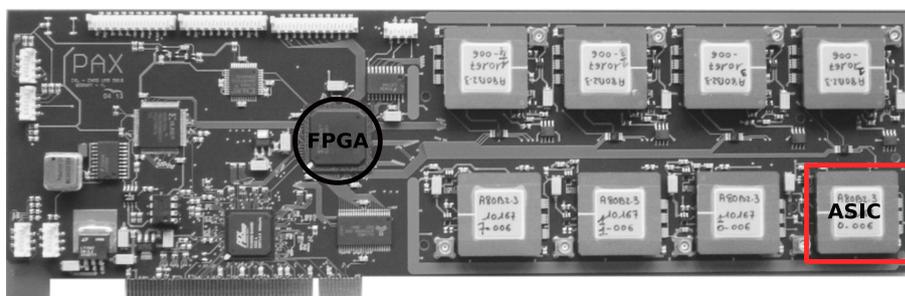


FIG. 2.1 – Photo du système PAX comprenant 8 ASICs (dont un entouré sur la droite) émulant les neurones et 1 FPGA (entouré sur la gauche) gérant la connectivité en réseau de neurones.

Neurones HH sur circuits analogiques (1/puce)

Les ASICs présentés ci-dessus ont été développés par l'équipe *Ingénierie des Systèmes Neuromorphiques* et permettent de reproduire le PA des

¹Application-Specific Integrated Circuit

²Field-Programmable Gate Array

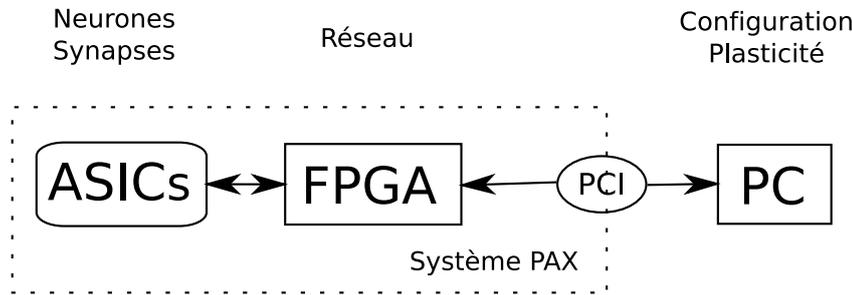


FIG. 2.2 – Organisation du système PAX. Les ASICs simulent les neurones et leurs synapses. Le FPGA gère le réseau. Le PC calcule la plasticité et communique au FPGA, via le connecteur PCI, les poids synaptiques mis à jour.

neurones en temps réel [70, 71, 72, 38, 73]. En effet, les ASICs calculent en analogique, en continu et respectent le temps réel biologique. Le modèle de neurone utilisé s'appuie sur le formalisme de Hodgkin et Huxley présenté au chapitre 1.

Le système permet donc la modélisation de réseaux du type *spiking neural network* avec un modèle complexe de neurones. L'avantage principal d'un tel système neuromorphique sur la simulation logicielle est la vitesse de fonctionnement. Un autre avantage est sa capacité de communiquer en analogique avec le vivant. Prenons l'exemple du mode de simulation présenté dans la section 2.3.1 qui a été reproduit avec le logiciel NEURON. Pour réaliser une simulation de six minutes biologiques, le système PAX cadencé sur le temps réel, nécessite six minutes réelles alors qu'avec le logiciel NEURON, il a fallu trois heures de simulation sur un ordinateur³.

Pour modéliser le potentiel de membrane du neurone, la contribution de quatre conductances ioniques voltage-dépendants est possible. Leurs courants respectifs sont I_{Na} , I_K , I_{fuite} et I_{mod} . Ceci permet de simuler quelques types particuliers de neurones corticaux comme nous le verrons un peu plus loin.

L'activité des synapses est également représentée par une conductance dite synaptique, ou encore multi-synapse, dont le courant est I_{syn} . Le circuit possède deux entrées : une inhibitrice et une excitatrice. Ces entrées synaptiques sont non seulement utilisées pour intégrer les messages nerveux

³Intel® Core(TM)2 CPU, 2,13GHz, cache : 2048 Ko, SDRAM : 3,5Go. Le système d'exploitation est le GNU/Linux Ubuntu® version 6.10.

d'autres neurones, mais également pour injecter du bruit synaptique. Nous avons vu dans le chapitre précédent que nous considérons le bruit stimulant chaque neurone comme étant un seul pattern de stimulations aléatoires représentant le grand nombre de stimulations afférentes.

Chaque ASIC émule un seul neurone et fournit le potentiel de membrane par une sortie analogique permettant sa visualisation. Il possède également une sortie numérique qui, grâce à un comparateur, permet de détecter les PA. Une entrée analogique de l'ASIC permet d'injecter un courant de stimulation continu. D'autres entrées permettent de configurer le circuit.

- Conductances ioniques

Nous avons vu au chapitre 1 le modèle de Hodgkin et Huxley dont nous allons rappeler les principes généraux. Rappelons que le potentiel de membrane est défini par la différence de potentiel entre l'intérieur et l'extérieur de la cellule et résulte du déplacement d'ions à travers des protéines transmembranaires, dits canaux ioniques. Ces canaux sont de différents types, sensibles chacun à une catégorie d'ions spécifiques. Ainsi le potentiel de membrane est dû à la contribution de différentes espèces ioniques.

Ajoutons que l'activité des canaux ioniques dépend du potentiel global de membrane. La capacité membranaire peut alors être assimilée à un condensateur et les conductances ioniques sont adjointes en parallèle. En incluant les conductances sodique, potassique, de fuite, du canal modulant (pour un neurone excitateur RS) et synaptique, nous obtenons le schéma électrique de la figure 2.3 où le potentiel aux bornes du condensateur représente le potentiel membranaire de la cellule nerveuse.

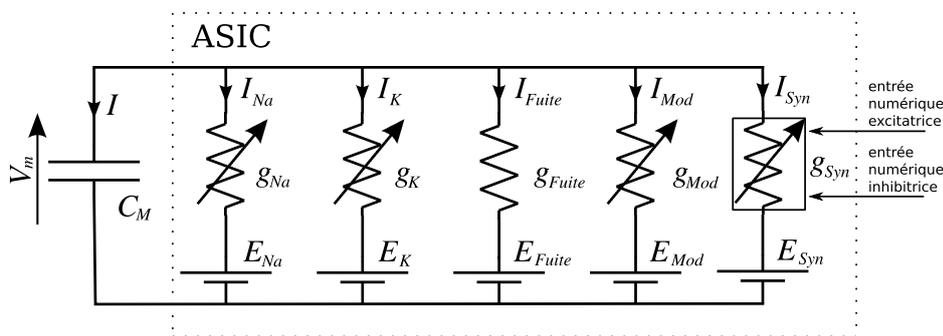


FIG. 2.3 – Schéma électrique du modèle de Hodgkin et Huxley avec des conductances sodique, potassique, de fuite, modulant et les synapses.

Chaque ASIC est connecté à un condensateur extérieur représentant la capacité de membrane C_M situé sur la carte PCB⁴. Il ne peut être intégré à l'intérieur de l'ASIC à cause de sa valeur de capacité trop élevée.

L'annexe A fournit les détails du modèle implanté (équations du modèle ainsi que valeurs des paramètres).

- Conductances synaptiques

Le modèle synaptique à base de conductances utilisé est le modèle de multi-synapse. Rappelons que le poids synaptique est représenté par un temps de montée qui détermine son amplitude. Ainsi, au niveau du circuit l'entrée synaptique numérique permet, par la durée de son activation, de déterminer la durée du créneau relatif au poids. Rappelons également que l'on parle de multi-synapse car le neurone ne possède qu'une unique entrée sur laquelle on peut cumuler plusieurs sorties provenant d'autres neurones ou simplement des stimulations synaptiques pouvant représenter du bruit.

- Caractéristiques fonctionnelles

D'un point de vue fonctionnel, les neurones présentent des caractéristiques comme les courbes de fréquences *versus* courant de stimulation (appelées courbes $f(I)$, voir figure 2.4), et l'adaptation de fréquence de décharge pour les neurones RS (figure 2.5), cohérentes avec les caractéristiques fonctionnelles de neurones correspondant au modèle. En effet, pour un neurone RS, le canal modulant fournit un effet d'adaptation de la fréquence de décharge du neurone lorsque celui-ci est stimulé un courant constant (représenté dans le circuit par un créneau de tension constante). Au début de la stimulation, la fréquence est élevée et diminue rapidement pour atteindre une fréquence relativement constante. Dans le cas du neurone de type FS, la fréquence est dès le début de la stimulation constante. Qu'il s'agisse des neurones RS ou FS, les courbes $f(I)$ sont variables d'un neurone à l'autre mais gardent dans l'ensemble la même forme. La figure 2.6 présente les différentes courbes $f(I)$ des neurones des différents neurones de PAX. Précisons que le choix des neurones dans le cadre de FACETS s'est porté sur les types RS et FS car il s'agit des neurones respectivement excitateur et inhibiteur les plus répandus dans le cortex cérébral.

⁴Printed Circuit Board

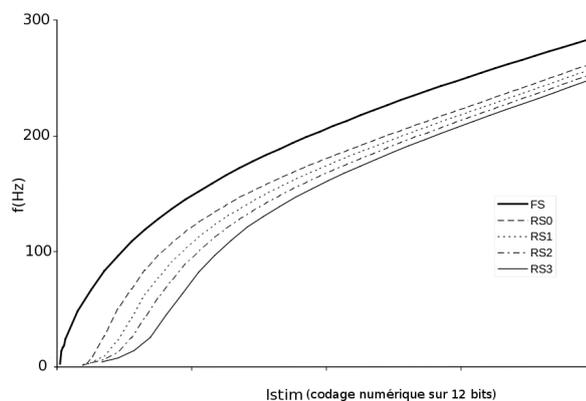


FIG. 2.4 – Exemple de courbes de fréquence stabilisée du neurone en fonction du courant de stimulation en entrée pour un neurone donné. Les valeurs du courant et la plage de valeurs peuvent varier d'un neurone à l'autre. FS correspond au neurone réglé en mode *Fast Spiking*. RSx correspondent à différentes valeurs de la conductance du modèle RS (RS0 : $g_{mod} = 45.5\mu S/cm^2$, RS1 : $g_{mod} = 90.9\mu S/cm^2$, RS2 : $g_{mod} = 136.8\mu S/cm^2$, RS3 : $g_{mod} = 181.8\mu S/cm^2$).

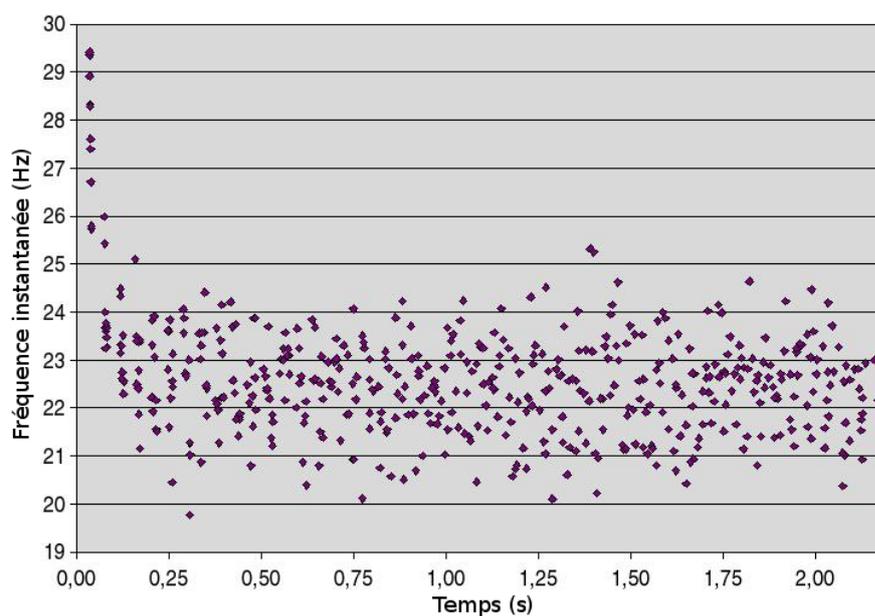


FIG. 2.5 – Exemple de réponse mettant en avant l'effet d'adaptation. En ordonnée nous avons la fréquence instantanée des PA d'un neurone en réponse à une stimulation en fonction du temps. Nous avons superposé les résultats de 10 essais sur un même neurone.

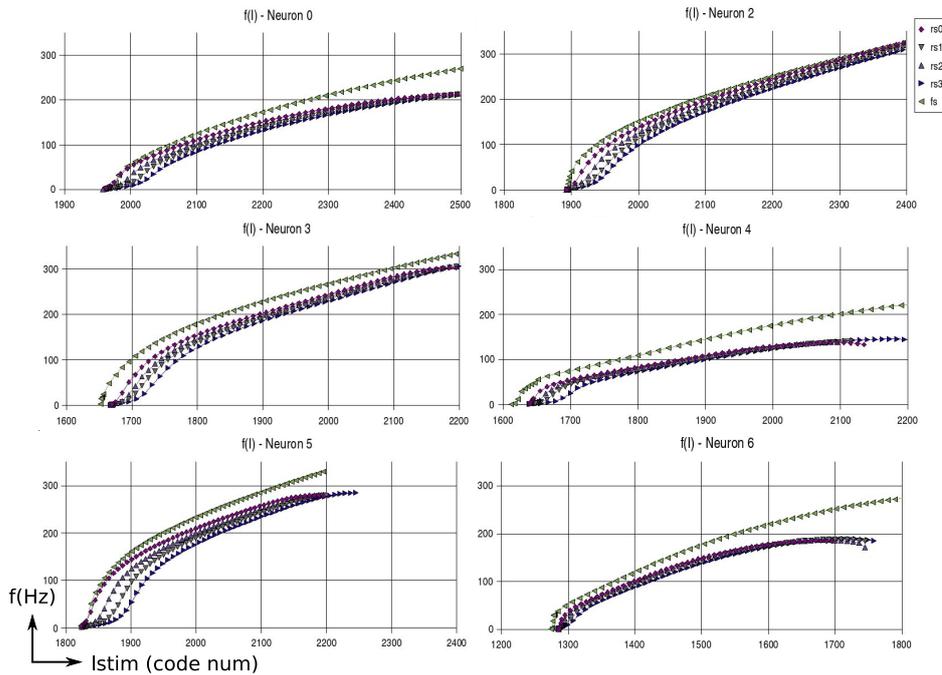


FIG. 2.6 – Ensemble des courbes $f(I)$ pour les 6 neurones utilisés. Les axes des ordonnées représentent la fréquence en Herz, et les abscisses le codage numérique de courant injecté. Pour plus de détails voir la figure 2.4

Gestion de la connectivité par FPGA

Les ASICs sont intégrés sur une carte PCB permettant le raccordement à un ordinateur via un bus PCI. Un circuit FPGA sur la carte permet de communiquer avec l'ordinateur, de piloter les circuits et ainsi d'établir la connectivité du réseau (figure 2.2). Le FPGA transforme une valeur numérique de poids en durée d'impulsion et l'applique à l'entrée synaptique du circuit concerné. De son côté, l'ordinateur envoie les paramètres de configuration du réseau de neurones. Il calcule également la plasticité et met à jour les valeurs des poids synaptiques stockés dans le FPGA⁵.

Durant ces travaux de thèse nous avons développé un programme en C qui calcule la plasticité synaptique. Le FPGA envoie les événements au programme qui calcule alors, en utilisant l'algorithme de STDP, la variation des conductances synaptiques concernées et les met à jour en les renvoyant au circuit intégré via le FPGA.

⁵A terme le calcul de plasticité sera reporté sur le FPGA, voir chapitre 3.

Description de l'implémentation de la STDP

Nous avons implémenté en langage C l'algorithme de STDP décrit au chapitre 1 en nous basant sur l'algorithme d'intégration détaillé en annexe B. Cet algorithme simplifie le calcul puisque l'équation 1.19 implique de prendre en considération tous les couples possibles de PAs pré et postsynaptiques. Un schéma de calcul itératif a alors été choisi. Dans le cas de la potentialisation, le calcul se fait au temps d'émission du PA postsynaptique (t_i) par le neurone i qui dépend des temps d'émission de tous les PAs (k de 0 à m) présynaptiques déjà émis (t_j^k) par le neurone j . Ce calcul est le suivant, selon l'équation 1.19 :

$$\sum_{k=0}^m P[t_i - t_j^k] \epsilon_i \epsilon_j^k \quad (2.1)$$

Cette somme peut-être calculée en introduisant un facteur de modulation S_j^m comme suit :

$$S_j^m = S_j^{m-1} \cdot \exp[-(t_j^m - t_j^{m-1})/\tau_p] + \epsilon_j^m \quad (2.2)$$

Ainsi le calcul est simplifié en mettant à jour uniquement le facteur de modulation à chaque PA.

De la même manière, dans le cas de la LTP, de l'équation 2.3, nous passons à un facteur de modulation S_i^m dont l'expression est fournie dans l'équation 2.4.

$$\sum_{k=0}^m Q[t_j - t_i^k] \epsilon_i^k \epsilon_j \quad (2.3)$$

$$S_i^m = S_i^{m-1} \cdot \exp[-(t_i^m - t_i^{m-1})/\tau_q] + \epsilon_i^m \quad (2.4)$$

Au final, tous les PAs, qu'ils soient considérés pré ou postsynaptiques, sont pris en compte en mettant à jour les facteurs de modulation.

2.1.2 Automatisation du système

L'ordinateur permet de superviser le fonctionnement du système comme nous l'avons déjà évoqué (figure 2.2). Nous avons développé un ensemble de programmes en langage C pour la communication et le calcul de plasticité ainsi que des scripts permettant l'automatisation du système (figure 2.7). L'annexe C présente les différents programmes, scripts de traitement et formats de fichiers.

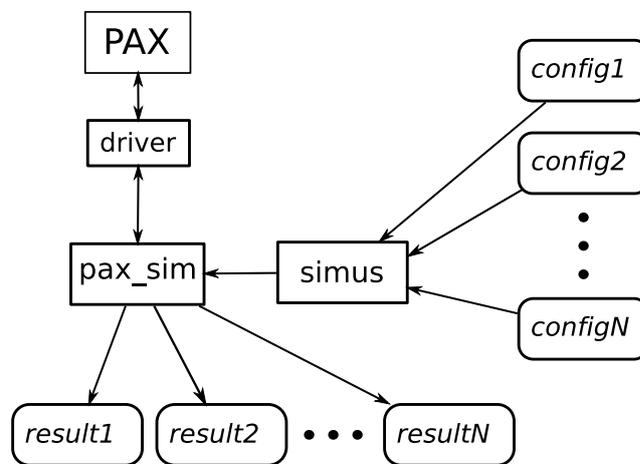


FIG. 2.7 – Organisation des programmes et données pour lancer des simulations. L'unité *simus* recherche des simulations à lancer et pour chacune (*configX*) il transfère au programme *pax_sim* les données nécessaires. Ce dernier se charge alors de la simulation en initialisant le système *PAX* (plus précisément le FPGA) à travers le *driver* et calcule la plasticité. Le programme génère des fichiers de résultat pour chaque simulation, ici représentés par *resultX*.

Au niveau informatique, un *driver* permet de communiquer avec le FPGA de la carte PCB via le protocole PCI. Il fournit les fonctions de base de plus bas niveau permettant d'envoyer et de recevoir des instructions sous forme de mots.

A un niveau plus élevé, nous avons développé des fonctions en langage C permettant un ensemble d'opérations telles que :

- initialiser le système en vidant toutes les mémoires
- créer une connexion entre deux neurones
- déterminer (ou modifier) les paramètres d'une connexion entre deux neurones (polarité, poids synaptique, délai axonal)
- appliquer un courant de stimulation à un neurone
- enregistrer un pattern de stimulations synaptiques (poids synaptique et ensemble d'intervalles de temps entre stimulations) auquel sera soumis un neurone
- détecter un potentiel d'action (lecture du numéro de neurone et du temps d'émission)
- démarrer une simulation (soumission des neurones aux patterns de stimu-

lations synaptiques enregistrés)
- arrêter la simulation

A un niveau encore supérieur, d'autres fonctions permettent de récupérer les données de la simulation que sont les temps d'émission de potentiels d'actions avec les numéros de neurones concernés. Puis à chaque événement ainsi détecté, le programme calcule pour chaque connexion liée au neurone concerné, la variation de plasticité suivant la règle de STDP et met à jour les poids des connexions modifiées. Ces données sont également sauvegardées.

Pour configurer le système avant une simulation, 4 fichiers de configuration en mode texte renseignent le courant de stimulation appliqué à chaque neurone, la connectivité ainsi que les poids initiaux et le pattern de bruit de chaque neurone.

Afin d'automatiser les simulations, un script se charge de récupérer tous les fichiers de configuration regroupés dans un même dossier, et pour chacun lance une simulation avant de passer au suivant une fois la simulation terminée. Ceci permet l'exécution d'un grand nombre de simulations sans intervention de l'utilisateur.

Une fois l'ensemble des simulations effectuées, un script permet de lancer des traitements qui vont hiérarchiser les résultats de simulation, effectuer des calculs et générer des graphes. Les traitements sont effectués par des programmes en langage Perl, Python et Gnuplot (voir annexe C).

2.1.3 Détermination des valeurs biophysiques manipulées

-Difficultés rencontrées

Les modèles de neurones étant implantés sur des circuits analogiques, les grandeurs manipulées présentent une imprécision absolue due aux défauts de fabrication. Il y a donc, à l'instar des systèmes biologiques, une intrinsèque variabilité dans le comportement des circuits. De plus la stimulation synaptique ainsi que la stimulation par injection de courant sont gérées par des valeurs numériques qui ne peuvent être liées directement à des valeurs biophysiques. La conversion directe est impossible d'une part à cause des dispersions électroniques des composants internes à l'ASIC mais aussi aux composants externes (comme la capacité de membrane), d'autre part à cause de l'impossibilité d'accéder aux paramètres des canaux ioniques. Par conséquent nous ne pouvons fournir les valeurs des paramètres biophysiques des

conditions de stimulation des neurones et des synapses en simulation.

-Solutions apportées

Nous pouvons, dans un premier temps, caractériser les circuits en fonction de leurs valeurs numériques de stimulation. Nous obtenons alors les courbes $f(I)$ pour chaque neurone (figure 2.4). Dans un second temps, connaissant le modèle exact implanté, il est possible de le simuler par une autre voie. Le logiciel NEURON a été utilisé pour reproduire l'activité du modèle de neurone utilisé dans les ASICs. La simulation avec NEURON donne l'avantage de pouvoir maîtriser les stimulations des neurones avec des valeurs des paramètres biophysiques. Ainsi, la simulation informatique nous permet d'obtenir une équivalence des valeurs numériques utilisées dans la stimulation des neurones électroniques.

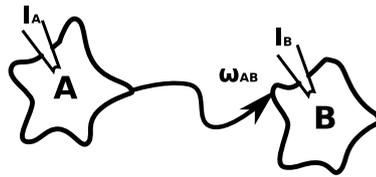


FIG. 2.8 – Réseau de 2 neurones A et B. ω_{AB} est la synapse connectant A à B. I_A et I_B sont des courants injectés respectivement à A et à B.

Correspondance des conductances synaptiques

Afin de déterminer la valeur de paramètres biophysiques correspondant à une valeur numérique nous avons utilisé la méthode suivante : On crée un réseau de 2 neurones A et B (figure 2.8). Sans connexion synaptique, on stimule chaque neurone indépendamment l'un de l'autre. Ainsi, que ce soit sur les modèles de neurones électroniques ou informatiques, on cherche le courant de stimulation permettant d'impliquer une fréquence particulière d'émission de PA du neurone. Le neurone A aura un courant I_A lui imposant une fréquence f_A d'émission de potentiels d'action. Le neurone B stimulé par le courant I_B aura une fréquence f_B . On règle I_A et I_B afin que $f_B < f_A$, et pour cela nous pouvons nous référer aux courbes $f(I)$. Puis, le principe est de créer une synapse excitatrice, d'efficacité w_{AB} , entre A (neurone présynaptique) et B (neurone postsynaptique). Le poids w_{AB} est au départ nul, puis il est incrémenté jusqu'à atteindre la valeur permettant de synchroniser le neurone B au

neurone A. En faisant une approximation linéaire des 2 courbes d'évolution de la fréquence en fonction du poids (résultats présentés à la figure 2.9), nous obtenons 2 droites (une pour la simulation sur neurones électronique et une logicielle). En faisant correspondre les deux équations des deux droites nous obtenons une équivalence de la valeur numérique utilisée avec les ASICs et la valeur biophysique utilisée avec NEURON. En utilisant deux neurones B et A respectivement cadencés à 3 et 8.5 Hz (fréquence légèrement variable pour la simulation matérielle), nous avons obtenu la loi de correspondance suivante : $w_{bio} = \frac{0,0943w_{num}+3,0562}{1070,33}$ où w_{bio} est la valeur biophysique en Siemens du poids synaptique équivalent à la valeur numérique de poids w_{num} . Ainsi, par cette technique, nous estimons que la conductance synaptique maximale utilisable par le système électronique est d'environ 20 nS.

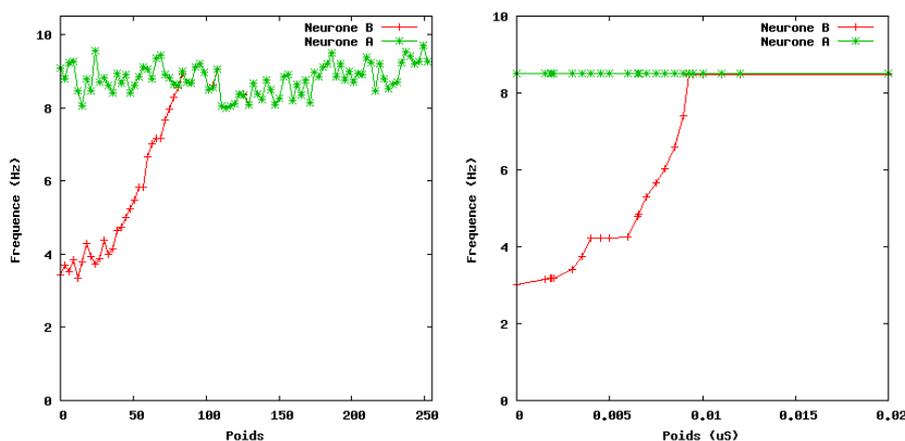


FIG. 2.9 – Gauche : simulation avec un réseau de 2 neurones électroniques. La courbe montre l'évolution de la fréquence du neurone B en fonction de la conductance synaptique (dont la valeur est numérique) entre A et B. Droite : même simulation avec le logiciel NEURON avec des valeurs biophysiques de conductance synaptique.

Correspondance des courants de stimulation

Le courant de stimulation est codé sur 12 bits. Une plage de valeurs sur 9 bits est suffisante pour déclencher des oscillations du neurone sur toute sa gamme de fréquences. Le décalage rajouté aux 9 bits est utilisé pour compenser les disparités des neurones électroniques. Une première méthode que nous avons voulu utiliser est la caractérisation par la courbe $f(I)$. Cependant, la sensibilité du neurone vis-à-vis du courant de stimulation est trop impor-

tante, ce qui ne nous permet pas de caractériser correctement la fonction de transfert pour les faibles fréquences d'oscillations. L'utilisation des plus hautes fréquences n'est pas non plus possible. Les segments de courbes $f(I)$, sans avoir leur origine, ne peuvent pas être mis en relation en raison de la différence de comportement d'un neurone à l'autre (figure 2.6). Nous avons alors décidé de mettre en oeuvre une seconde méthode qui s'inspire de celle décrite pour la caractérisation des conductances synaptiques. Le principe de la méthode consiste à exciter un neurone B à travers une synapse afférente depuis le neurone A. La stimulation appliquée au neurone A (courant I_A) est constante, et, par conséquent, sa fréquence aussi (exactement 8.5 Hz pour la simulation logicielle et environ 8.5 Hz pour le matériel, voir figure 2.10). Le poids de la synapse est également fixe. Par contre, le courant de stimulation de B (I_B) est variable. Il évolue de façon régulière à partir d'une valeur qui ne permet pas au neurone l'émission de PAs, malgré la présence de la synapse de A vers B, jusqu'à une valeur permettant au neurone B d'osciller en étant synchronisé avec le neurone A. La même méthode est utilisée avec le logiciel NEURON. Les données recueillies pour chaque valeur du courant de stimulation I_B du neurone B sont la fréquence de B f_B , celle du neurone A f_A (en principe constante) et le courant de stimulation I_B de B. On règle I_A et I_B afin que $f_B < f_A$. Pour cela nous pouvons nous référer aux courbes $f(I)$. La correspondance du poids fixé sur le matériel et sur NEURON a été obtenue grâce à la loi de correspondance établie précédemment. Là encore, nous n'avons pas exactement une correspondance linéaire entre la fréquence et le courant de stimulation. Cependant la loi de correspondance des courants de stimulation, à l'instar de la loi de correspondance des poids, donne un ordre de grandeur et aide à approximer la valeur fournie au système matériel. Nous obtenons par exemple pour le neurone dont il est question sur la figure 2.10, la loi de correspondance du courant de stimulation, appliquée au neurone, suivante : $I_{bio} = -0,01625.I_{num} + 36,363$ où I_{bio} est la valeur biophysique en nA du courant équivalent à la valeur numérique I_{num} appliquée au neurone électronique.

2.2 Préparation de l'étude et de l'analyse

Avant de passer aux études réalisées, nous allons présenter dans cette partie la méthode de corrélation du bruit synaptique qui sera utilisée pour

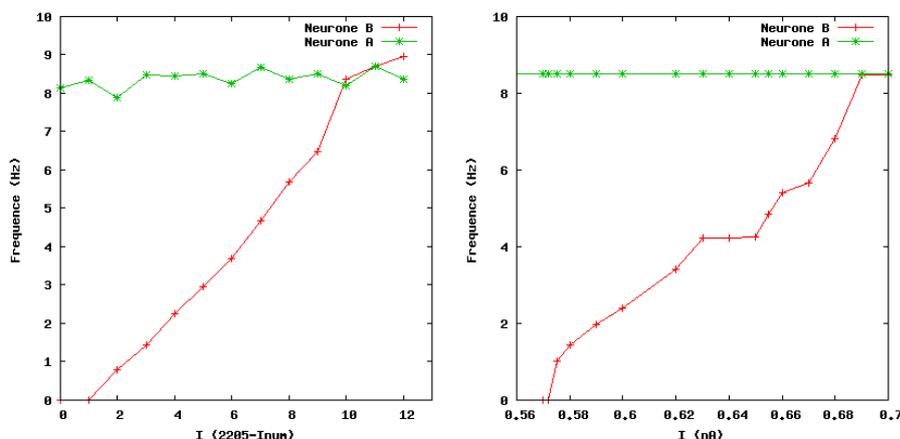


FIG. 2.10 – Gauche : simulation avec un réseau de 2 neurones électroniques A et B dont la conductance synaptique est fixe. La courbe montre l'évolution de la fréquence du neurone B en fonction du courant de stimulation appliqué à B. Droite : même simulation avec le logiciel NEURON avec des valeurs biophysiques de courant de stimulation.

générer les stimuli d'entrée des neurones, ainsi que les outils qui nous permettrons d'analyser les résultats que nous obtiendrons.

2.2.1 Corrélation des patterns de bruit

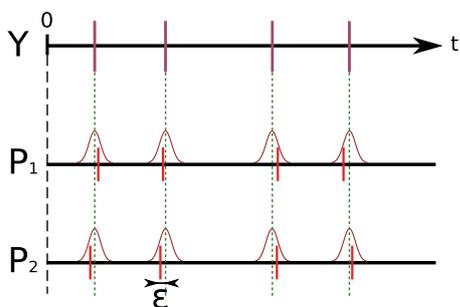


FIG. 2.11 – Le pattern d'évènements Y suit une loi de Poisson. Les patterns de bruit P_i sont corrélés à partir de Y .

Chaque neurone est stimulé par un pattern d'entrée qui lui est propre et qui représente le bruit généré par les nombreuses connexions afférentes d'un neurone cortical. Un neurone isolé stimulé par un pattern de bruit dont la fréquence moyenne des stimuli est de 5 Hz, aura une fréquence de décharge d'environ 3 Hz.

Les patterns de bruit appliqués en entrée des neurones sont générés selon

deux paramètres variables : la fréquence des évènements du pattern et la corrélation entre les différents patterns.

Il n'existe pas de méthode standard pour corréler de tels patterns et nous n'avons pas trouvé de méthode satisfaisante pour le contrôle fin que l'on désire. Par conséquent, nous avons développé la méthode décrite ici.

Afin d'obtenir des patterns de bruit corrélés entre eux, nous utilisons donc une distribution de Poisson initiale à partir de laquelle les patterns corrélés seront générés. Cette méthode de corrélation nous permet de maîtriser finement le degré de corrélation mais nous éloigne des caractéristiques d'une loi de Poisson.

Une séquence X d'intervalles de temps x_i est générée suivant une loi de Poisson comme suit :

$$X = \{x_1, \dots, x_n\} / x_i = N(0, 1) \cdot \sqrt{m - 1/2} + m \quad (2.5)$$

où $N(0, 1)$ est une loi normale, m l'intervalle moyen entre deux stimulations. La variable X est ensuite convertie en un pattern de temps absolu Y :

$$Y : y_i = \sum_{j=1}^i x_j \quad (2.6)$$

Les évènements des patterns de bruit sont alors générés autour de chaque élément de Y . L'écart temporel entre l'évènement de Y et celui d'un pattern de bruit est donné par la formule suivante :

$$\varepsilon = \frac{N(0, 1)}{3} \cdot (1 - \alpha) \cdot \frac{T}{2} \quad (2.7)$$

où T la période moyenne et $\alpha \in [0, 1]$ est le coefficient de corrélation des patterns de bruit. $N(0, 1)$ est une loi normale et $N(0, 1)/3$ permet d'avoir 99,7% du tirage entre -1 et 1. Si le tirage est hors de l'intervalle $] - 1, 1[$, il est ignoré et réitéré jusqu'à obtenir un tirage valide.

2.2.2 Outils d'analyse

Histogramme de poids

Les données recueillies après une simulation fournissent le temps auquel le poids d'une synapse a été recalculé et sa nouvelle valeur, ainsi que les nu-

méros de neurone présynaptique et postsynaptique. Afin d'évaluer la convergence des poids synaptiques, nous superposons dans un premier temps toutes les valeurs de poids sur un même graphique tout au long de la simulation (graphique A figure 2.12, graphiques A et B de la figure 2.19).

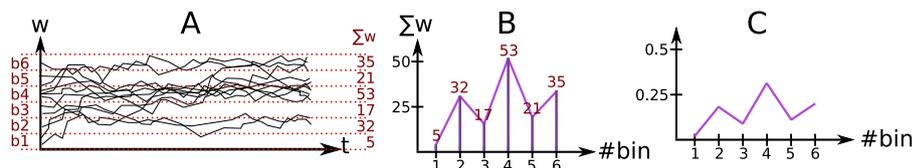


FIG. 2.12 – Construction de l'histogramme de poids synaptiques. w est le poids synaptique, t le temps, $\#bin$ le numéro de section [1-6] (pour les sections de b1 à b6), $\sum w$ le compte du nombre de valeurs de poids.

Puis nous réalisons un histogramme qui nous permet de visualiser la répartition des poids synaptiques sur la durée d'une simulation de manière plus synthétique. Le graphique est obtenu en divisant l'axe des poids synaptiques en petites portions du poids total (figure 2.12 A). La taille de section qui nous a paru adéquate au vu des résultats obtenus est d'environ 2% du poids maximal. Par conséquent, nous obtenons un découpage en 36 sections pour les données issues des simulations. Le numéro de section est relatif à la valeur des poids. Ainsi plus le numéro est élevé, plus les valeurs de poids qu'elle représente l'est aussi.

Pour chaque section, nous effectuons le décompte des valeurs de poids s'y trouvant. Ceci correspond au nombre de calculs de poids par la STDP ayant donné une valeur dans cet intervalle. Nous avons alors des informations sur la présence plus ou moins importante de synapses ayant eu, au long de la simulation, une valeur de poids qui variait à l'intérieur de cette plage de valeurs. Dans la figure 2.12, le nombre de valeurs de poids de chaque section a été reporté à droite dans le graphique A ($\sum w$). Le graphique B est obtenu en calculant le nombre de valeurs de poids en fonction du numéro de section. Puis la répartition des poids est générée par normalisation du graphique B dans le graphique C. Le graphique construit par cette méthode fournit alors la distribution normalisée en ordonnée en fonction du numéro de la section de poids en abscisse (voir les graphiques du haut de la figure 2.21). Ainsi si la répartition (graphique C) tend vers 0, ceci montrera une distribution des poids avec des valeurs faibles. Inversement, si la répartition augmente, ceci montrera une distribution des poids avec des valeurs élevées.

Catégories de synapses

Dans la deuxième étude réalisée avec PAX que nous présenterons dans la prochaine section, deux groupes de neurones sont distingués. Ainsi, nous considérerons 4 catégories de synapses relativement aux 2 groupes de neurones (figure 2.13). Les synapses reliant les neurones du *Groupe 1* aux neurones du *Groupe 2* sont appelées *synapses sortantes* ; celles connectant les neurones du *Groupe 2* à celles du *Groupe 1* sont appelées *synapses entrantes*. Les synapses connectant les neurones à l'intérieur du *Groupe 1* sont appelées *synapses internes*, et celles à l'intérieur du *Groupe 2*, *synapses externes*.

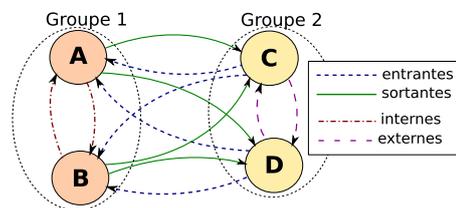


FIG. 2.13 – Catégories de synapses dépendant du groupe d'appartenance.

Représentation poids multiples avec pondération de Pearson VII

Dans l'étude concernant l'évolution des poids des 4 catégories de synapses, 4 principaux types de convergence ont été observés. La distribution des poids peut être répartie près des valeurs maximales, près des valeurs minimales, les 2 à la fois (répartition bimodale), ou confinée autour d'une plage de valeurs (unimodale). Afin de pouvoir représenter ces 4 catégories par un seul point sur un graphique, nous avons besoin de deux coordonnées que nous appellerons pF pour poids forts et pf pour poids faibles. Autrement dit, la coordonnée de pF caractérisera la quantité de synapses ayant un poids fort, et celle de pf la quantité de synapses ayant un poids faible. Ces deux coordonnées seront créées en pondérant deux fois les histogrammes de poids présentés en section 2.2.2. La première pondération renforcera l'influence des poids forts, ce qui donnera la valeur de pF , et la seconde pondération renforcera l'influence des poids faibles pour donner la valeur de pf . Nous avons utilisé dans un premier temps des pondérations linéaires mais cela n'a permis la discrimination que des 3 premières distributions présentées ci-dessus. Il a donc fallu trouver une fonction qui conserve les poids de valeurs moyennes tout en renforçant les valeurs extrêmes. En testant plusieurs fonctions répon-

nant aux exigences suscitées, il en ressort que la fonction la plus adéquate ressemble à une demi-gaussienne avec une excroissance en son centre. Les fonctions de type Pearson VII (éq. 2.8) permettent d'approximer des gaussiennes et comportent plusieurs paramètres permettant d'affiner avec une grande précision la forme de la fonction.

$$f(x) = \left[O_1 + K_1^2 \frac{(x - x_{01})^2}{M_1} \right]^{-M_1} + \left[O_2 + K_2^2 \frac{(x - x_{02})^2}{M_2} \right]^{-M_2} \quad (2.8)$$

où O_i , K_i , x_{0i} , M_i ($i \in \{1, 2\}$) sont les paramètres et x la variable de la fonction f .

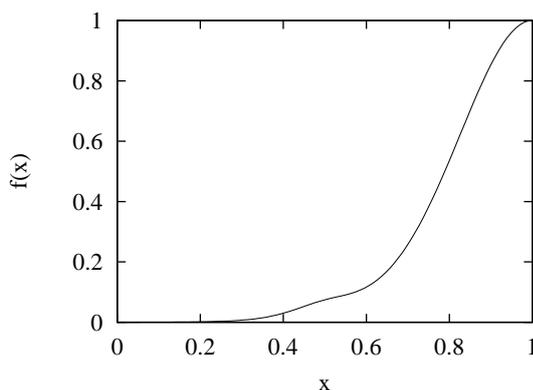


FIG. 2.14 – Tracé de la fonction $f(x)$ de l'équation 2.8 avec les valeurs de paramètres du tableau 2.1.

Le tracé de la fonction $f(x)$ (figure 2.14) avec l'histogramme de poids normalisé sur $[0,1]$, nous donne une répartition des poids favorables au poids forts, donc la valeur de pF . La fonction $f(1-x)$ avec le même histogramme normalisé donnera la valeur de pf . Une exploration automatisée, sur de larges plages de valeurs pour chaque paramètre de l'équation, a fourni une classification optimale avec les valeurs du tableau 2.1. La discrimination des 4 types de convergence est donc possible en fonction des coordonnées fournies par les 2 paramètres pF et pf comme présenté à la figure 2.15.

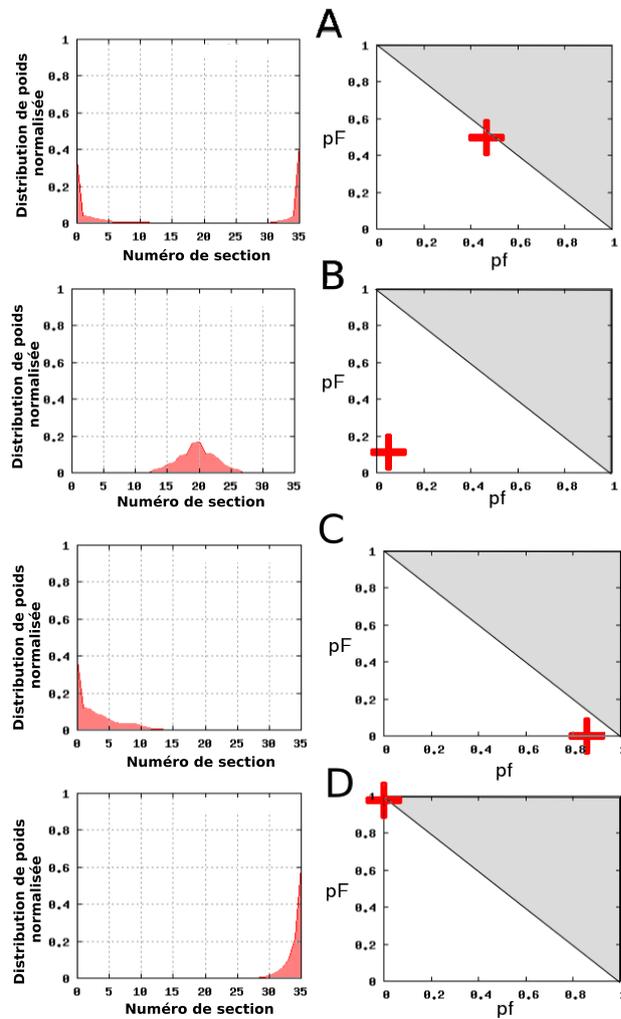


FIG. 2.15 – Pour chaque groupe de graphiques, à gauche nous avons l'histogramme de répartition des poids normalisé et à droite le graphique pF vs pf . *A* : convergence bimodale, $(x; f(x))$ tend vers $(0,5; 0,5)$. *B* : convergence confinée, $(x; f(x))$ tend vers $(0; 0)$. *C* : convergence vers des valeurs minimales, $(x, f(x))$ tend vers $(1; 0)$. *D* : convergence vers des valeurs maximales, $(x; f(x))$ tend vers $(0; 1)$.

i	O_i	K_i	M_i	x_{0_i}
1	1	4	15	1
2	2,8	15	3	0,5

TAB. 2.1 – Paramètres optimaux de l'équation 2.8 pour discriminer des répartition bimodales, confinées, maximales et minimales.

Histogramme de corrélation

Afin d'évaluer la corrélation entre les patterns de PAs émis par les neurones, nous construisons un histogramme à partir du diagramme de PAs avec le temps en abscisse et le numéro de neurone en ordonnée (graphique A figure 2.16).

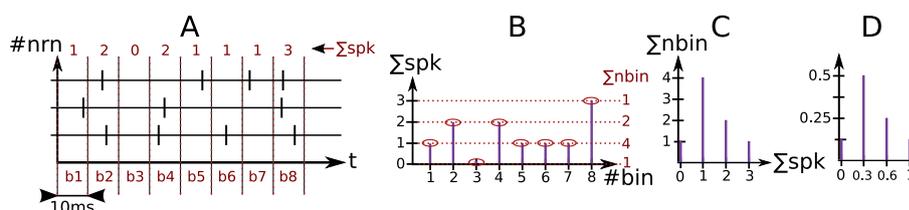


FIG. 2.16 – Construction du diagramme de corrélation. $\#nrn$ est le numéro de neurone, $\#bin$ est le numéro de section [1-8] (pour les sections de b1 à b8), $\sum spk$ est le compte du nombre de PAs d'une section, $\sum nbin$ est le compte du nombre de sections ayant un compte de PAs particulier.

La méthode consiste à découper l'axe temporel en sections de 10 ms chacune. Dans chaque section, on dénombre les PAs émis (graphique B). Le nombre maximum correspond au total de neurones du réseau (6 dans le système mais 3 dans l'illustration) à condition que dans une fenêtre de 10 ms un neurone en particulier n'émette qu'un seul PA. Dans nos simulations, l'activité des neurones ne dépasse pas des fréquences de 10 Hz, ce qui correspond à une période moyenne de 100 ms et rend peu probable l'apparition de deux potentiels d'action dans une fenêtre de 10 ms pour un même neurone.

Pour chaque nombre de PAs possibles (1 à 6 dans le système et 1 à 3 dans l'illustration), on compte le nombre de sections correspondant. Le nombre possible de PAs est mis en abscisse et le nombre de sections en ordonnée (graphique C). Ces 2 axes sont ensuite normalisés (graphique D).

Nous obtenons par conséquent un graphique montrant la distribution

2.3. EFFET DE LA CORRÉLATION DU BRUIT SUR LA PLASTICITÉ 55

des PAs relative à une fenêtre temporelle minimale (10 ms). Si les PAs des différents neurones sont corrélés dans un tel laps de temps, la distribution tendra vers 1, alors que si les PAs des neurones sont peu corrélés, ils n'appartiendront pas aux mêmes sections temporelles et la distribution tendra alors vers 0.

Pour avoir une idée de la corrélation globale, on effectue une somme pondérée linéaire du graphique D. Cette somme pondérée caractérisera la corrélation globale de la simulation.

2.3 Etudes de l'effet de la corrélation du bruit sur la plasticité

Nous allons présenter dans cette partie les deux études réalisées ainsi que les résultats et analyses respectifs de ces études.

2.3.1 Corrélation identique du bruit synaptique attaquant les neurones

Etude réalisée

L'objectif de cette étude est d'évaluer l'effet de la corrélation des patterns de bruit appliqués en entrée des neurones d'une part sur l'évolution de la distribution des valeurs des conductances synaptiques dont les synapses sont soumises à la STDP, et d'autre part sur la corrélation des patterns de sortie des neurones.

-Simulation matérielle avec PAX

Nous réalisons des simulations pour 4 différentes fréquences d'attaque de bruit : 5, 8, 10 et 15 Hz afin de montrer l'influence de la fréquence sur la corrélation des réponses des neurones.

Pour chaque fréquence de bruit, tous les neurones sont stimulés par des patterns de bruits corrélés de manière identique. Onze corrélations différentes (avec un coefficient de corrélation variant de 0 à 1) ont été testées. Dans chaque cas, les poids synaptiques initiaux ont été initialisés avec différentes configurations : tous nuls, tous maximaux, aléatoires (3 patterns différents). Une simulation dure 6 minutes, ce qui permet d'avoir un réseau stabilisé sur une longue partie de la simulation (au bout de quelques dizaines de secondes).

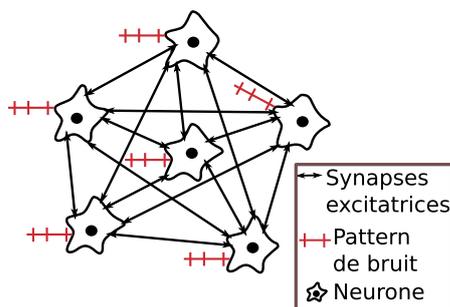


FIG. 2.17 – Réseau de 6 neurones excitateurs inter-connectés et stimulés par des patterns de bruit corrélés. Les neurones sont inter-connectés par des synapses excitatrices soumises à un algorithme de STDP.

Ainsi 220 simulations ont été réalisées. Sur un tel exemple les propriétés du système en temps réel sont particulièrement appréciées.

Durant la simulation, l'algorithme de STDP influence l'évolution des poids synaptiques reliant les 6 neurones inter-connectés en *all-to-all* (figure 2.17).

Ajoutons que les neurones sont stimulés par des courants constants compris entre 0,4 et 0,5 nA, valeurs biologiques déterminées par la méthode de la section 2.1.3. Avec ces réglages, les neurones soumis à un pattern de bruit de 5 Hz, oscillent avec une fréquence d'environ 3 Hz, ce qui est conforme au contexte d'observations de la plasticité de type STDP dans les systèmes biologiques [60]. Nous avons représenté sur la figure 2.18 la fréquence moyenne d'oscillation de chaque neurone calculée en divisant le nombre de PAs par le temps de simulation. La fréquence n'étant pas stable, nous avons calculé la fréquence instantanée pour chaque PA (par rapport au PA précédent) ainsi que la moyenne de ces fréquences instantanées. Nous avons représentés ces calculs dans la figure 2.18(a) par la moyenne des fréquences instantanées par neurone et leur écart-type. Les valeurs des conductances synaptiques varient entre 0 et 20 nS. Ce réglage évite les comportements de *bursting*, c'est-à-dire des trains de PAs à fréquence beaucoup plus importante que la fréquence moyenne. Rappelons enfin que lorsqu'un neurone émet un potentiel d'action, toutes les conductances synaptiques lui étant liées sont recalculées avec l'algorithme de STDP.

-Simulation logicielle avec NEURON

Nous avons réalisé certaines des simulations avec NEURON. Rappelons que PAX, à cause de la variabilité de ses composants, fournit une activité qui

2.3. EFFET DE LA CORRÉLATION DU BRUIT SUR LA PLASTICITÉ 57

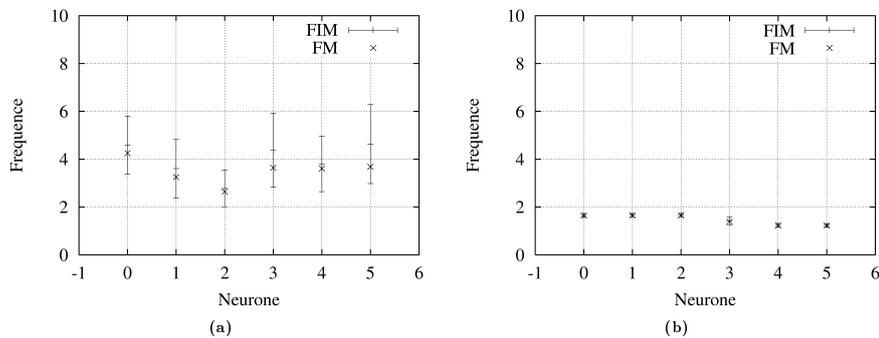


FIG. 2.18 – Fréquences moyennes (FM) et Fréquences instantanées moyennes (FIM) de décharge, avec l'écart-type, de chaque neurone soumis à un bruit dont la fréquence est de 5 Hz pour PAX sur le graphique de gauche et pour NEURON sur le graphique de droite.

n'est pas exacte par rapport au modèle simulé contrairement à NEURON. Nous avons donc réalisé des simulations avec NEURON afin de pouvoir les comparer avec celles réalisées avec PAX. Afin d'être le plus proche possible des conditions du système PAX, nous avons utilisé les valeurs calculées et cités ci-dessus pour les courants appliqués aux neurones ainsi que les poids synaptiques initiaux. Par conséquent nous avons appliqué aux neurones des courants de stimulation dont l'équivalent des valeurs biophysiques de l'amplitude est comprise entre 0,4 et 0,5 nA et des conductances synaptiques entre 0 et 20 nS. Avec les réglages effectués, un neurone soumis à un bruit de 5 Hz présente une activité dont la fréquence est autour de 1,5 à 2 Hz comme on peut le voir sur le graphique de la figure 2.18(b). De plus, la décharge des neurones se fait avec des légers *bursts* (séquence de 2 à 4 PAs séparés d'environ 10 ms) accentués si l'on augmente le niveau de la stimulation.

Les conditions entre NEURON et PAX ne sont pas parfaitement identiques mais suffisamment proches pour les phénomènes observés par notre étude.

Résultats et analyse

Evolution des poids synaptiques

Les résultats suivants correspondent aux résultats des simulations dont la fréquence du bruit est de 5 Hz. Les résultats des simulations, conformément au type de plasticité, présentent des poids qui varient en permanence au gré

des PAs émis (A et B de la figure 2.19).

On observe ici avec une corrélation des patterns de bruit en entrée des neurones nulle ($\alpha = 0$, α de l'équation 2.7), que les poids ont une tendance à évoluer à l'intérieur d'une plage de valeurs restreinte. La simulation avec NEURON a entraîné une évolution à l'intérieur d'une plage plus restreinte qu'avec la simulation sur PAX. Les graphiques A et B présentent l'évolution brute des poids pour respectivement PAX et NEURON. Les graphiques C et D montrent bien que pour les deux cas nous avons une répartition autour d'une plage de valeurs. Enfin le graphique E confirme que cette répartition, conformément à ce que l'on voit sur A, B, C et D, est plus restreinte pour NEURON que pour PAX.

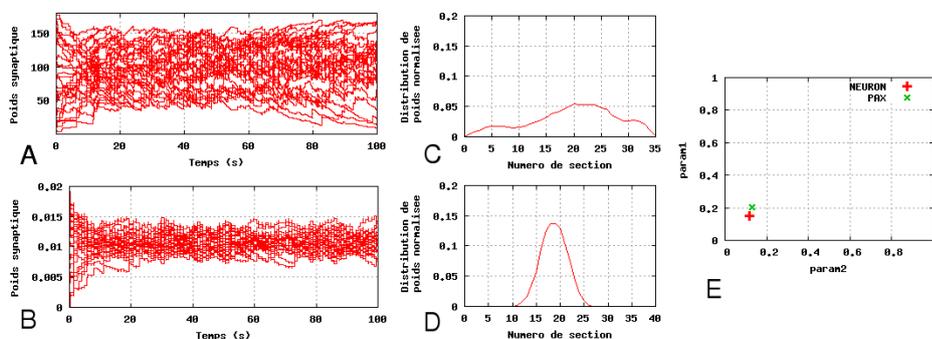


FIG. 2.19 – Résultats de simulation, de durée 100 secondes, avec une corrélation du bruit où $\alpha = 0$ pour PAX (A, C, E) et NEURON (B, D, E). A, B L'évolution des poids bruts. C, D Histogrammes de poids de cette évolution. E Représentation de ces évolutions sur un même graphique avec la méthode de pondération de Pearson (N pour NEURON et P pour PAX).

Pour la simulation avec PAX, au fur et à mesure que la corrélation d'entrée augmente, la répartition des poids devient mixte (poids dans les extrêmes et dans une plage de valeurs). Ceci est illustré par la figure 2.20 pour les corrélations dont α est compris entre 0,4 et 0,8. Ce phénomène n'est observé avec NEURON que pour $\alpha = 0,8$ mais pas lorsque $\alpha = 0,5$. Cela étant aucune valeur de α entre 0,5 et 0,8 n'a été testée.

Pour les fortes valeurs de corrélation, la répartition devient quasi bimodale (une partie des poids ont des valeurs maximales et l'autre minimales) avec NEURON ainsi qu'avec PAX comme on peut le voir sur les deux graphiques de la figure 2.20 (points très proches de $(0,5; 0,5)$). Remarquez que

2.3. EFFET DE LA CORRÉLATION DU BRUIT SUR LA PLASTICITÉ 59

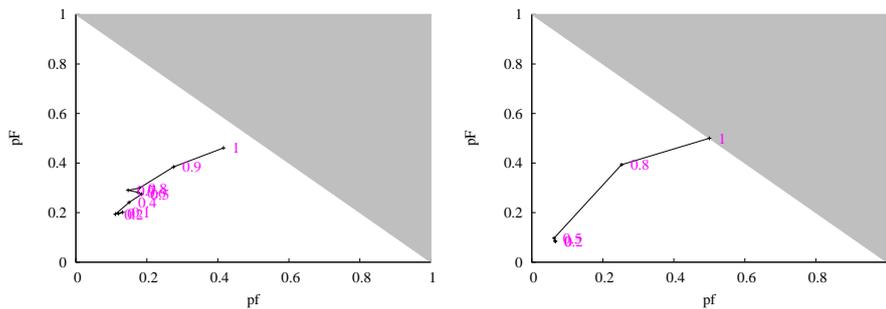


FIG. 2.20 – Evolution du mode de répartition des poids en fonction de la corrélation du bruit de fréquence 5 Hz. *A gauche* Simulation avec PAX. Les points reliés du plus proche de l'origine au plus éloigné correspondent aux valeurs respectives de α 0 ; 0,2 ; 0,4 ; 0,5 ; 0,6 ; 0,7 ; 0,8 ; 0,9 ; 1. *A droite* Simulation avec NEURON. Les points reliés du plus proche de l'origine au plus éloigné correspondent aux valeurs respectives de α 0 ; 0,2 ; 0,5 ; 0,8 ; 1

NEURON donne une répartition bimodale presque parfaite.

La répartition des poids passe d'unimodale autour d'une plage de valeur à mixte entre unimodale et bimodale et enfin bimodale, suivant l'augmentation de la corrélation. On peut le voir plus précisément sur la figure 2.21.

En faisant varier la fréquence du bruit de 5 à 15 Hz, la figure 2.22 montre que la tendance est toujours la même à savoir un passage en suivant la corrélation du bruit d'une répartition plutôt unimodale à une répartition qui devient exclusivement bimodale lorsque la corrélation du bruit est maximale.

Evolution de la corrélation de l'activité des neurones

La corrélation de l'activité des neurones augmente progressivement (figure 2.21). On observe cependant que la progression de la corrélation des sorties augmente brusquement quand $\alpha > 0,8$ comme on peut le voir sur les graphiques de la figure 2.23 pour les simulations avec PAX et NEURON. Nous voyons plus précisément la brusque augmentation sur les graphiques de corrélation de la figure 2.21.

Notons que la corrélation est plus élevée au niveau de l'activité des neurones simulés avec le logiciel NEURON. Nous l'expliquerons dans le paragraphe suivant.

Lorsque l'on augmente la fréquence du bruit, l'évolution de la corrélation des PAs est modifiée en étant plus faible pour les valeurs faibles de α et augmente brusquement quand α est proche de 1 (figure 2.24).

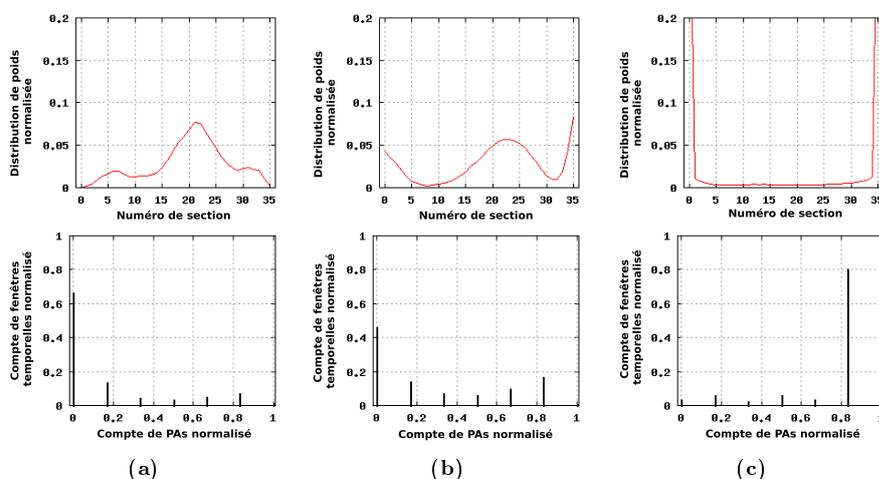


FIG. 2.21 – Simulation de l’effet de la corrélation du bruit d’entrée sur la convergence des poids synaptiques et la corrélation de l’activité des neurones. *Ligne du haut* : histogrammes de poids pour des simulations où la corrélation d’entrée α de l’équation 2.7 est 0,2 (a), 0,6 (b) et 1 (c). *Ligne du bas* : Histogrammes de corrélation pour des simulation où la corrélation α est 0,2 (a), 0,6 (b) et 1 (c).

Interprétations

Nous avons vu que l’on passe progressivement d’une répartition des poids confinée (unimodale) à une répartition bimodale lorsque la corrélation des patterns de bruit d’entrée augmente. Cependant l’état différencié bimodal n’est atteint que lorsque la corrélation du bruit est très forte. De plus l’état mixte semble atteint, avec le logiciel NEURON, que lorsque la corrélation du bruit est déjà forte alors que cet état est présent avec des corrélations moins importantes avec PAX. Bien que l’on puisse retrouver une même tendance, nous voyons une différence entre les simulations avec NEURON et avec PAX. Ce qui différencie les deux environnements de simulation est la présence d’un bruit intrinsèque aux neurones électroniques. Nous l’avons observé sur l’écart-type des fréquences d’émissions de PAs qui est assez important avec PAX et très faible avec NEURON. Les neurones biologiques sont également bruités et ce bruit intrinsèque au milieu confère des propriétés importantes dans l’état du cerveau. Avec ce système nous n’avons pas le contrôle sur le bruit généré mais cela nous place néanmoins dans un contexte voisin du contexte biologique. Il aurait été intéressant d’ajouter artificiellement au

2.3. EFFET DE LA CORRÉLATION DU BRUIT SUR LA PLASTICITÉ

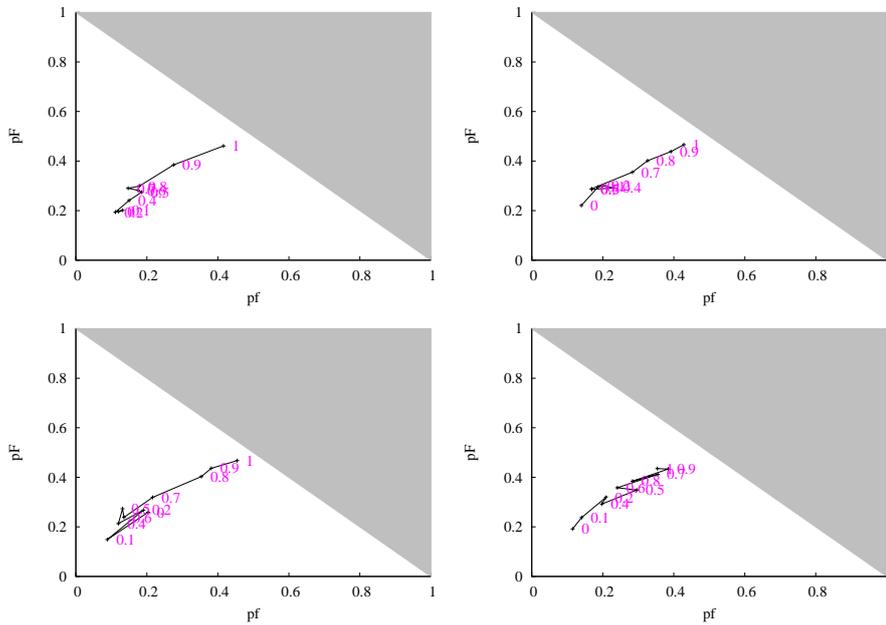


FIG. 2.22 – Evolution du mode de répartition des poids en fonction de la corrélation du bruit des simulations réalisées avec PAX avec une fréquence de bruit d'entrée variant de gauche à droite et de haut en bas respectivement de 5, 8, 10 et 15 Hz.

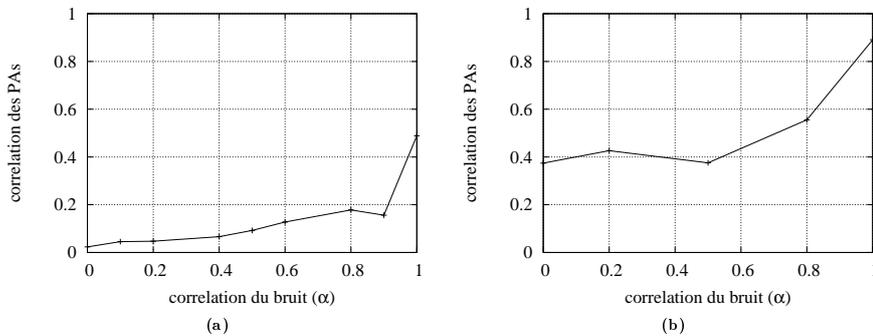


FIG. 2.23 – Evolution de la corrélation de l'activité des neurones en fonction de la corrélation du bruit d'entrée des simulations réalisées avec PAX (a) et avec NEURON (b). La fréquence du bruit est de 5 Hz.

modèle NEURON un bruit perturbant l'activité mais cela dépasse le cadre de notre étude.

Les études réalisées, explorant la relation entre corrélation d'entrées et

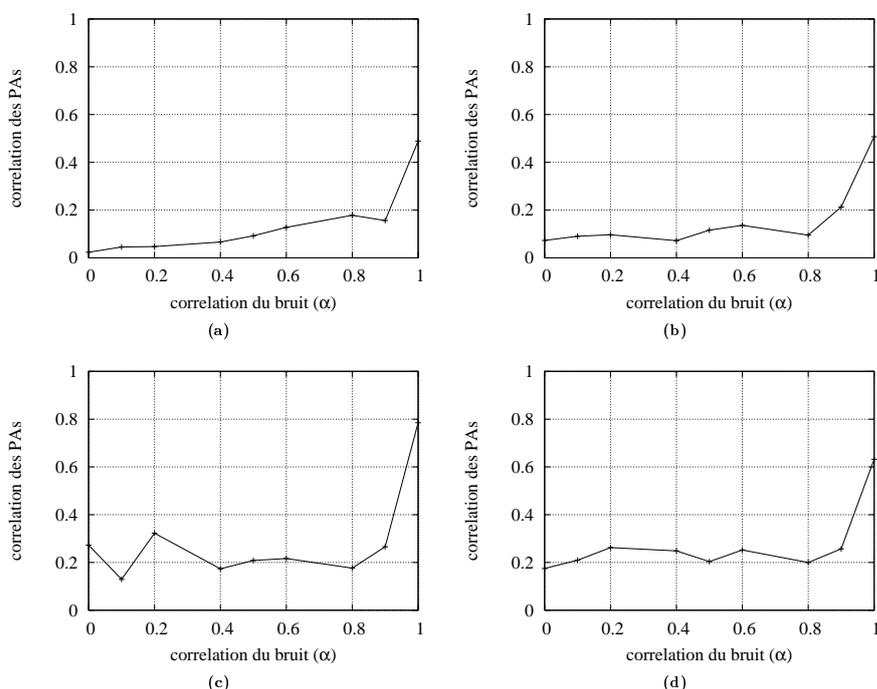


FIG. 2.24 – Courbes d'évolution de la corrélation de l'activité des neurones en fonction de la corrélation du bruit d'entrée des simulations réalisées avec PAX avec une fréquence de bruit d'entrée variant de 5 Hz pour (a), 8 Hz pour (b), 10 Hz pour (c) et 15 Hz pour (d).

STDP, ont montré que l'effet de la STDP était bien influencée par la corrélation. La plupart des modèles de STDP utilisés ne prennent pas en compte l'influence non-linéaire des PAs mise en avant par les triplets de PAs⁶ [55] que nous incluons à travers le facteur de suppression (voir l'équation 1.18 du premier chapitre) [58, 74, 50, 75]. Ces travaux ont été réalisés avec un modèle de STDP et un modèle de neurone de type I&F, et seuls les travaux de Song et al. 2000 et 2001 utilisent des *hard bounds*⁷, alors que les autres utilisent des *soft bounds*⁸ comme dans le modèle que nous utilisons. La convergence des poids synaptiques dans le cas de l'utilisation de *hard bounds* impliquent une distribution des poids bimodale contrairement aux modèles utilisant des

⁶Modulation du changement du poids synaptique due à l'interaction de couples successifs de PAs pré-/postsynaptiques.

⁷Valeurs limites du poids synaptique en-deçà et au-delà desquelles le poids est fixé à cette valeur limite.

⁸Valeurs limites du poids vers lesquels il tend de manière asymptotique.

2.3. EFFET DE LA CORRÉLATION DU BRUIT SUR LA PLASTICITÉ

soft bounds dont la distribution des poids synaptiques après stabilisation de l'apprentissage se fait de façon unimodale centrée autour d'une plage de valeurs. Dans les travaux de Zou et Destexhe 2007 [76] qui utilisent un modèle de neurone se basant sur le formalisme de Hodgkin et Huxley et sur le même modèle de STDP que nous (intégrant les facteurs de suppression et les *soft bounds*), la convergence est également unimodale. Dans tous les travaux suscités, le contexte de simulation est différent du nôtre. Leurs études se font avec un neurone possédant un grand nombre d'entrées présynaptiques (entre 100 et 1000). Ces entrées sont caractérisées par des patterns qui peuvent être corrélés et sont soumis à STDP. Ainsi, ces patterns représentent l'ensemble des entrées que peut recevoir un neurone alors que dans notre cas nous présentons toutes ces entrées par un seul pattern que l'on a appelé pattern de bruit et qui ne possède pas de plasticité. Nous avons focalisé notre étude sur les synapses soumises à STDP reliant entre eux les neurones d'un petit réseau. Ainsi la corrélation que nous avons étudiée se fait entre les patterns que reçoivent les différents neurones et non entre des patterns d'entrée d'un même neurone. Dans ce contexte particulier, nous avons pu observer une convergence plus complexe induisant une distribution à la fois bimodale et unimodale en fonction de la corrélation des patterns de bruit. Précisons que dans les travaux de Zou et Destexhe, la corrélation se fait sur deux groupes de patterns distincts que reçoit le neurone, et l'inversion de leur corrélation a pour effet de décaler la moyenne des valeurs de poids confinés.

Nous avons observé qu'avec le système PAX simulant un modèle biophysique de neurones dont les synapses sont soumises à un modèle biophysique de STDP, l'évolution des poids synaptiques se fait en mêlant deux types de convergence : un unimodal et un bimodal sans exclusivité à part pour une très forte corrélation des patterns de bruit d'entrée.

Nous allons étendre maintenant notre étude à un réseau dont les neurones sont séparés en deux groupes dont la corrélation est distincte pour chacun. On s'intéressera alors à l'influence de la corrélation sur les synapses à l'intérieur d'un groupe ainsi que celles allant/venant vers/de l'autre groupe.

2.3.2 Corrélation dans 2 groupes de neurones

Etude réalisée

Dans cette étude, le réseau utilisé est toujours composé de 6 neurones tous inter-connectés. Cependant, cette fois-ci les neurones sont répartis en 2 groupes en fonction de la corrélation du pattern de bruit auquel ils sont soumis. Autrement dit, chaque groupe de neurones recevra des patterns de bruit corrélés indépendamment l'un de l'autre (figure 2.25). Les réglages (courants de stimulation et valeurs des poids synaptiques) des neurones ont été faits de la même façon que dans l'étude précédente.

Nous ferons varier le nombre de neurones dans chaque groupe (entre 2 et 4). Nous testerons également différentes corrélations du *Groupe 1* et du *Groupe 2* avec $\alpha_{Groupe1} \geq \alpha_{Groupe2}$ (α de l'équation 2.7). Les valeurs de corrélation appliquées pour le premier groupe seront $\alpha_{Groupe1} = \{0; 0, 2; 0, 4; 0, 5; 0, 6; 0, 7; 0, 8; 0, 85; 0, 9; 0, 95; 1\}$ et pour le second $\alpha_{Groupe2} = \{0; 0, 2; 0, 4; 0, 5; 0, 7; 0, 8; 0, 9; 1\}$. Nous inter-changerons aussi les neurones entre les 2 groupes afin d'observer l'effet éventuel de la variabilité des neurones électro- niques (3 arrangements). Enfin, nous testerons 2 ensembles de poids initiaux aléatoires. Au total nous avons près de 1000 simulations avec une durée de 6 minutes pour chacune, ce qui correspond à près de 100 heures de simulation en temps réel. Les données enregistrées à chaque émission de PA, sont le temps d'émission, le numéro du neurone concerné, ainsi que tous les poids synaptiques de ses synapses afférentes et efférentes recalculés par la STDP.

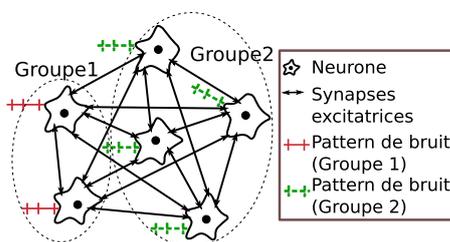


FIG. 2.25 – Réseau de 6 neurones excitateurs inter-connectés stimulés par deux groupes de patterns de bruit corrélés. Les neurones sont inter-connectés par des synapses excitatrices soumises à un algorithme de STDP.

2.3. EFFET DE LA CORRÉLATION DU BRUIT SUR LA PLASTICITÉ⁶⁵

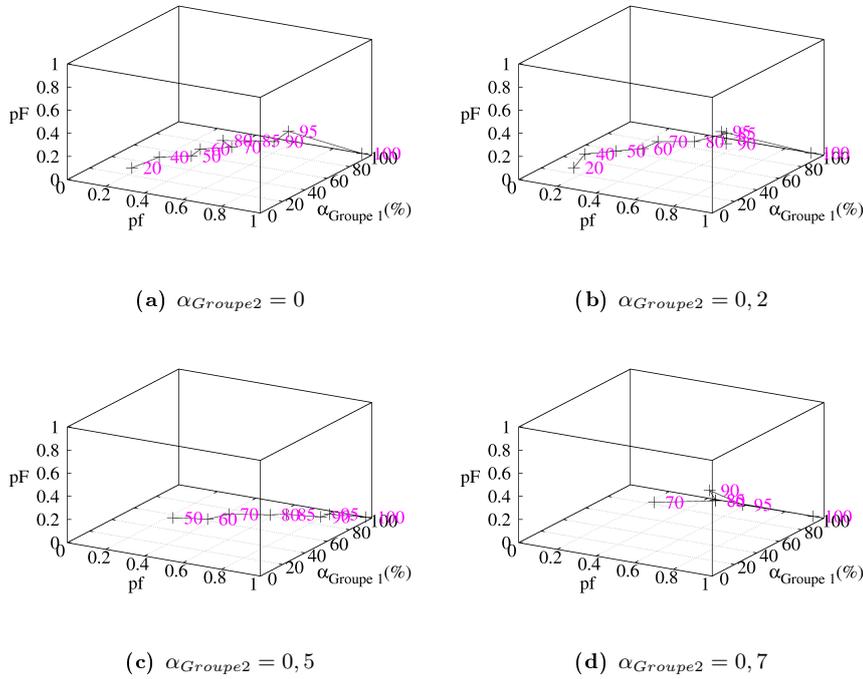


FIG. 2.26 – Synapses entrantes : répartition des poids synaptiques du réseau avec 4 neurones dans le Groupe 1 et 2 neurones dans le Groupe 2 en fonction de α_{Groupe1} et α_{Groupe2} .

Résultats et analyse

Pour plus de lisibilité, les résultats représentant les modes de répartition des poids sont présentés pour certains dans des graphiques en 3-dimensions. Aux graphiques 2-dimensions de la figure 2.15, une troisième dimension a été ajoutée représentant la corrélation du bruit (α) qui était uniquement représentée par une étiquette dans les résultats de l'étude précédente (figures 2.20 et 2.22). Notez que nous représentons α dans les graphiques en 3D par un pourcentage plutôt que par sa valeur comprise entre 0 et 1, et ce dans un souci de bien distinguer α des paramètres pF et pf dont les valeurs sont comprises également entre 0 et 1. Les figures de 2.26 à 2.33 présentent quelques résultats représentatifs de l'ensemble des 56 graphiques totaux obtenus suite à la simulation. Les commentaires ci-après illustrent les résultats obtenus et s'appuient sur ces graphiques.

Pour les corrélations des patterns de bruit du *Groupe 1* où $\alpha_{\text{Groupe1}} \geq$

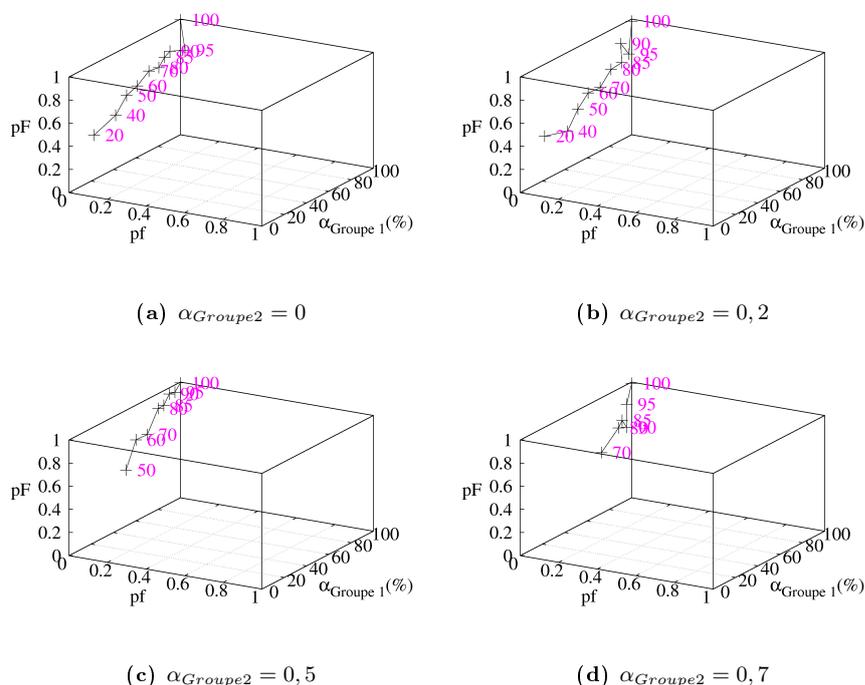


FIG. 2.27 – Synapses sortantes : répartition des poids synaptiques du réseau avec 4 neurones dans le Groupe 1 et 2 neurones dans le Groupe 2 en fonction de $\alpha_{Groupe1}$ et $\alpha_{Groupe2}$.

0,7, et quand le nombre de neurones du *Groupe 1* est plus grand ou égal (figures non présentées ici) à celui du *Groupe 2*, la convergence est indépendante de $\alpha_{Groupe2}$ pour les synapses entrantes et les poids sont proches des valeurs minimales (figure 2.26). Les poids des synapses sortantes sont au contraire proches des valeurs maximales (figure 2.27). Lorsque le nombre de neurones du Groupe 2 est plus grand (figure 2.28), alors il peut y avoir des convergences inversées (poids fort pour les synapses entrantes et faibles pour les synapses sortantes).

Lorsque $\alpha_{Groupe1} < 0,7$, les valeurs sont toujours réparties vers le même extrême mais en étant dans des plages de valeurs un peu plus faibles (figures 2.26 et 2.27).

Si l'on considère maintenant les synapses intérieures (appartenant au Groupe 1) et extérieures (appartenant au Groupe 2), nous constatons que la répartition a tendance à être mixte entre unimodale et bimodale bien qu'il soit difficile de tirer des conclusions plus précises, avec pour des valeurs de

2.3. EFFET DE LA CORRÉLATION DU BRUIT SUR LA PLASTICITÉ⁶⁷

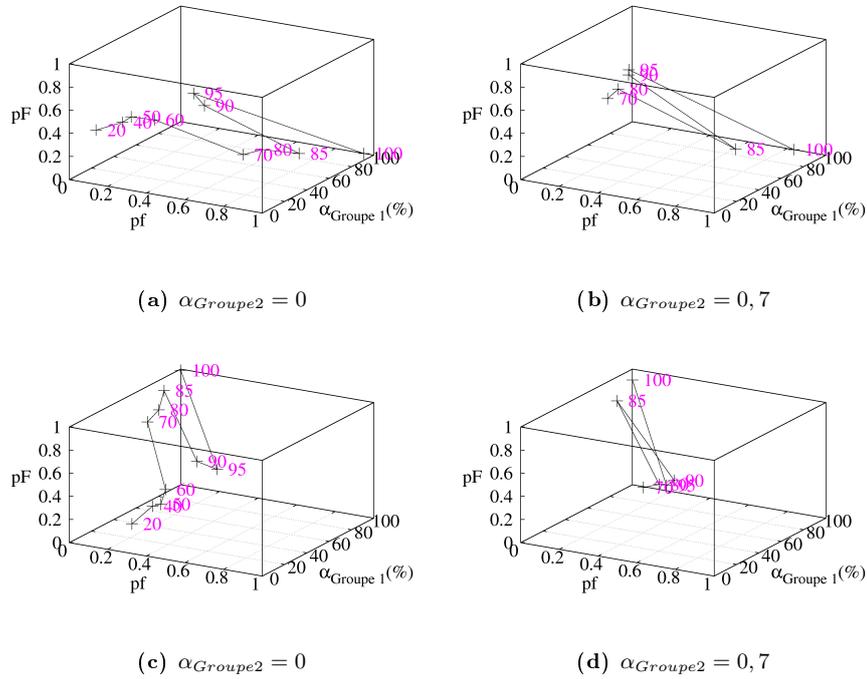


FIG. 2.28 – Répartition des poids synaptiques du réseau où le Groupe 1 comporte deux neurones et le Groupe 2 quatre. *En haut.* Graphiques de synapses entrantes. *En bas.* Graphiques de synapses sortantes.

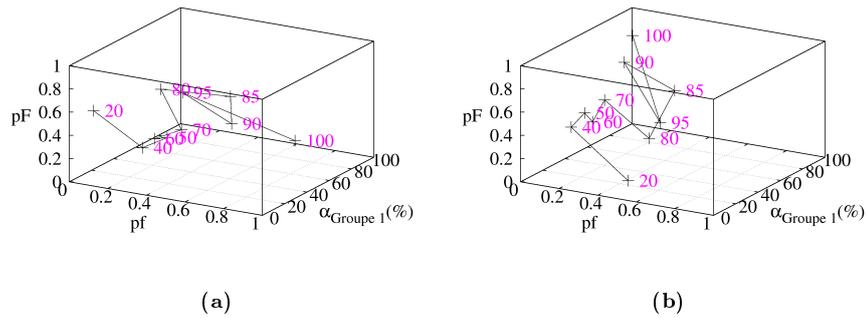


FIG. 2.29 – Synapses entrantes (a) et sortantes (b) : répartition des poids synaptiques du réseau avec $\alpha_{Groupe2} = 0,2$. Ici des neurones ont été permutés aléatoirement entre les groupes et sont donc différents de ceux des figures 2.26 et 2.27.

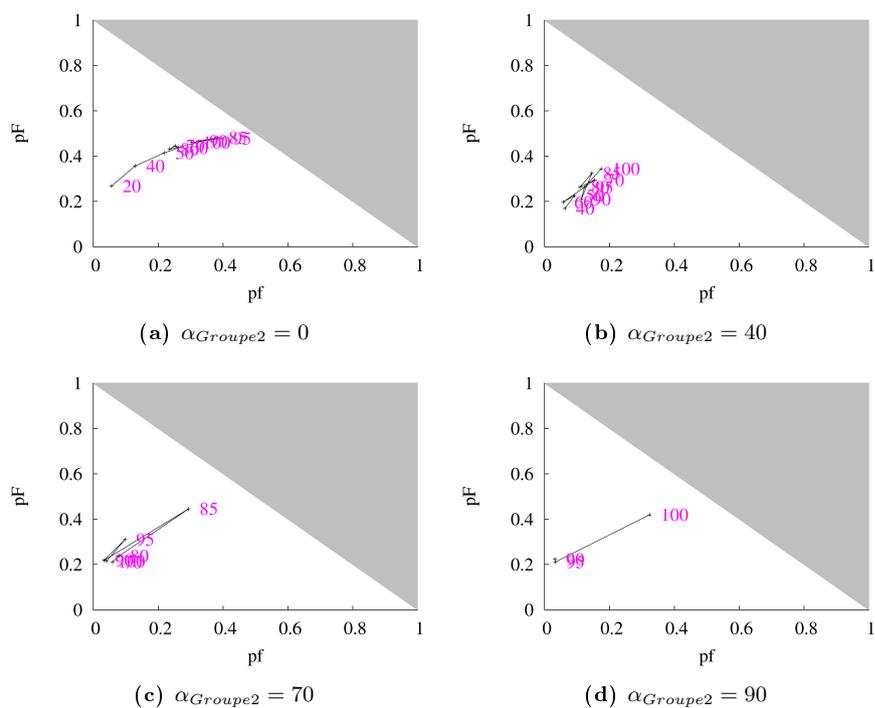


FIG. 2.30 – Synapses extérieures : répartition des poids synaptiques du réseau avec 4 neurones dans le Groupe 1 et 2 neurones dans le Groupe 2.

corrélation faible une répartition à tendance unimodale. Nous nous contentons de renvoyer aux figures de 2.30 à 2.33 qui montrent bien cette tendance avec parfois des petits écarts non généralisables.

Dans l'ensemble, l'état initial des poids synaptiques n'a pas eu d'effet significatif sur les observations que l'on a présenté ci-dessus. De plus, que le Groupe 1 possède 3 ou 4 neurones n'a pas eu d'influence remarquable.

Dans cette étude, nous avons considéré des neurones soumis à deux groupes indépendants d'activités synaptiques aléatoires corrélées avec une STDP appliquée entre tous les neurones. Quand la corrélation du Groupe 1 et son nombre de neurones sont supérieurs à ceux du Groupe 2, la STDP mène à une distribution des poids vers les valeurs minimales des synapses entrantes (du Groupe 2 vers le Groupe 1), et vers les valeurs maximales des synapses sortantes. Ceci montre le rôle moteur du groupe ayant le plus de neurones et la plus forte corrélation de ses entrées. A l'intérieur des groupes, nous avons remarqué que lorsque la corrélation des entrées est forte, les

2.3. EFFET DE LA CORRÉLATION DU BRUIT SUR LA PLASTICITÉ 69

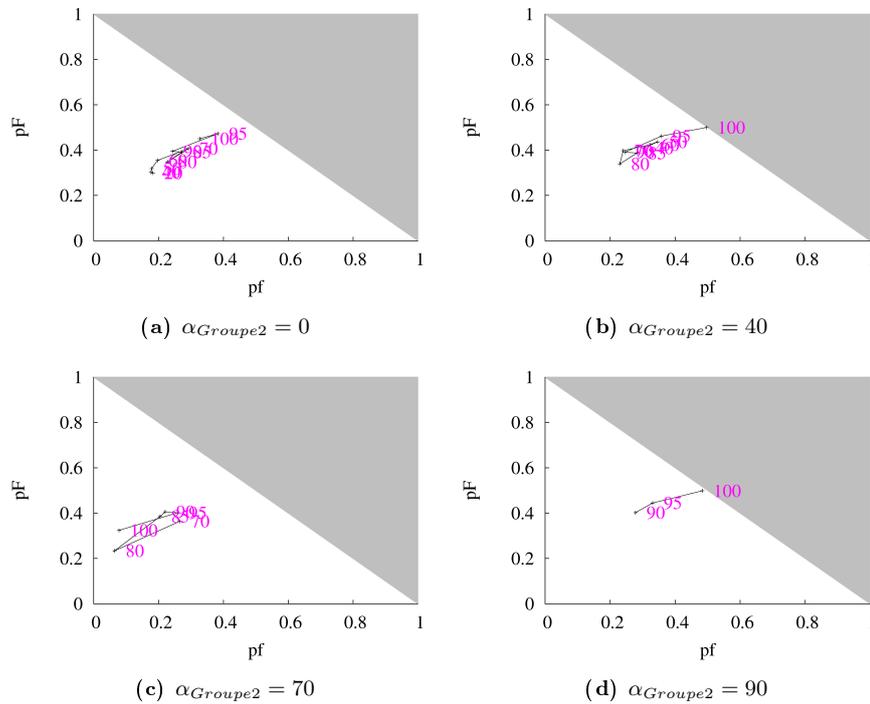


FIG. 2.31 – Synapses intérieures : répartition des poids synaptiques du réseau avec 4 neurones dans le Groupe 1 et 2 neurones dans le Groupe 2.

synapses sont discriminées et convergent vers une des valeurs extrêmes. À l'inverse, quand cette corrélation est faible, les synapses ont une distribution des poids unimodale.

Pour résumer les observations des résultats de ces simulations, nous notons qu'il ressort que lorsque la corrélation du Groupe 1 augmente, dans le cas des synapses entrantes, la répartition des poids passe d'unimodale à répartie vers les poids faibles. Dans le cas des synapses sortantes, la répartition passe d'unimodale à répartie vers les poids forts. Concernant les synapses à l'intérieur des groupes, la répartition passe d'unimodale à bimodale. L'unimodalité est observable uniquement lorsque la corrélation est suffisamment faible ($\alpha \leq 0,5$). Autrement dit, on observe que la convergence des poids synaptiques à l'intérieur d'un groupe semble indépendante de la corrélation de l'autre groupe. Seules les synapses inter-groupes sont affectées par la corrélation des deux groupes.

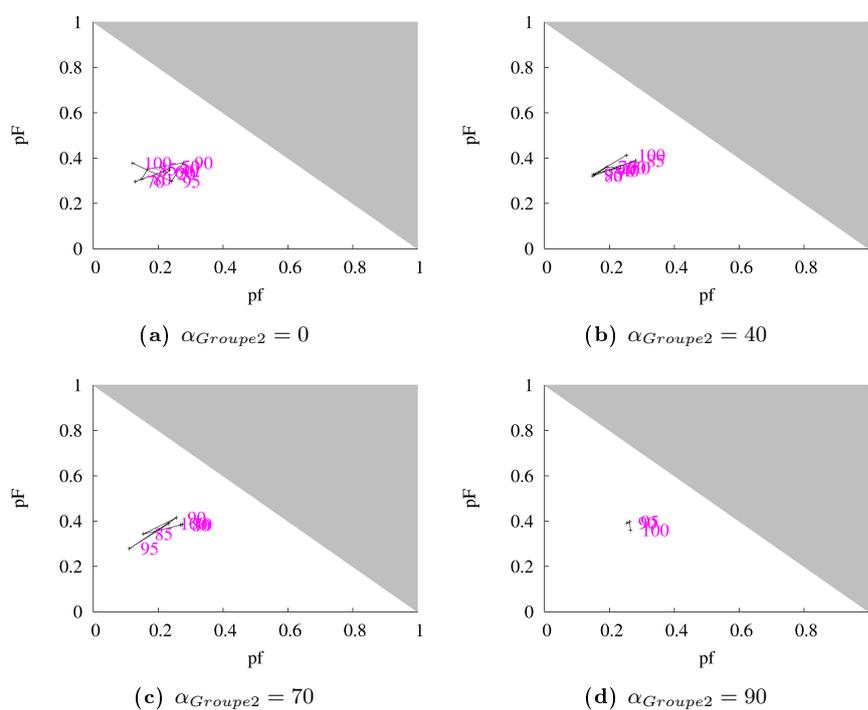


FIG. 2.32 – Synapses extérieures : répartition des poids synaptiques du réseau avec 2 neurones dans le Groupe 1 et 4 neurones dans le Groupe 2.

2.3.3 Conclusion

Nous avons étudié dans ce chapitre l'effet de la corrélation du bruit synaptique dans un petit réseau de neurones excitateurs inter-connectés. Ces études ont été réalisées en utilisant un système de simulation neuromorphique PAX et ont apporté des résultats nouveaux.

La première étude réalisée sur un réseau inter-connecté a permis de mettre en évidence que la corrélation du bruit influence le résultat du calcul de la STDP. Quand la corrélation augmente, la répartition des valeurs des poids synaptiques, initialement unimodale, devient bimodale. L'étude sur l'inter-connexion de groupes de neurones dont les entrées sont corrélées distinctement, permet d'observer l'influence de la corrélation entre ces différents groupes. Un groupe devient le meneur de tout le réseau, autrement dit ses synapses efférentes se renforcent et ses synapses afférentes se déprécient, à deux conditions : que la corrélation de ses patterns d'entrée de bruit soit supérieure à celle de l'autre groupe et que le nombre de neurones qui le

2.3. EFFET DE LA CORRÉLATION DU BRUIT SUR LA PLASTICITÉ 71

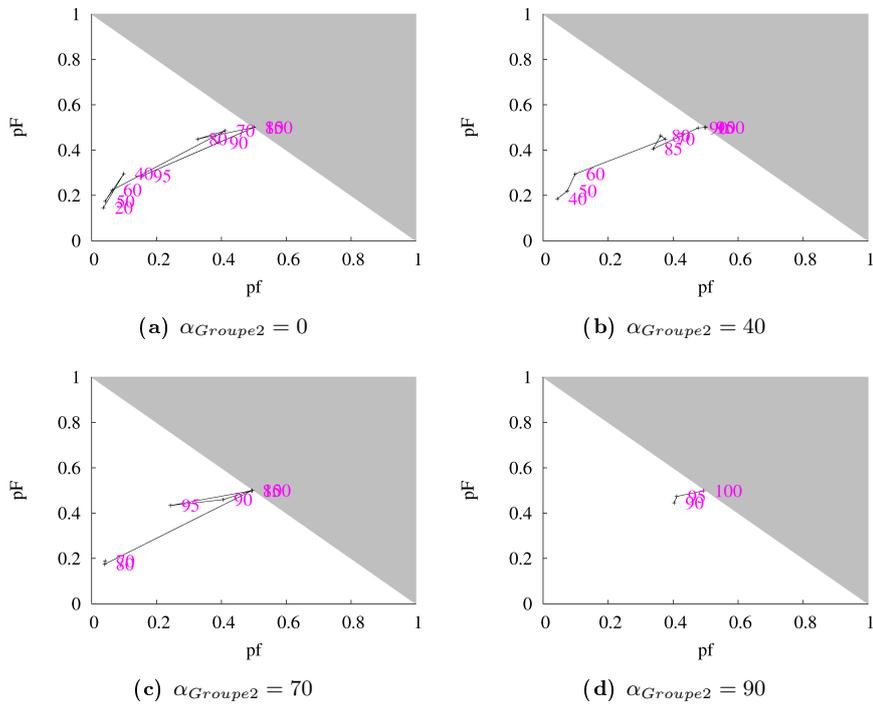


FIG. 2.33 – Synapses intérieures : répartition des poids synaptiques du réseau avec 2 neurones dans le Groupe 1 et 4 neurones dans le Groupe 2.

composent soit supérieur ou égal au second groupe.

Nous allons voir dans le prochain chapitre comment prendre en main un autre système neuromorphique offrant une plus grande flexibilité de configuration, et ce dans le but de poursuivre l'étude de la plasticité au sein de petits réseaux de neurones. Nous verrons également comment rendre le système accessible via Internet à la communauté scientifique en respectant un langage spécifique de description des simulations.

Chapitre 3

Environnement neuromorphique Gaillimh

Dans ce chapitre nous présentons le nouveau système de simulation, nommé Gaillimh, qui accueille plus de neurones artificiels et offre une plus grande liberté de configuration.

Après la présentation des caractéristiques du système, nous aborderons la méthodologie de configuration des circuits neuromorphiques pour les adapter à un modèle de neurone biologique.

Puis nous verrons les possibilités d'utilisation du système à travers les outils développés, mais aussi ses limitations.

Enfin, nous présenterons le travail réalisé afin de rendre accessible ce système (et les suivants) à la communauté scientifique avec la possibilité d'utiliser des scripts en PyNN¹, langage de programmation Python dédié à la simulation neuromorphique.

3.1 Système de simulation électronique Gaillimh

3.1.1 Propriétés du système et plasticité

- Organisation globale du système

Dans le chapitre précédent, nous avons réalisé une étude utilisant le système de simulation PAX qui permettait de gérer la mise en réseau des neurones simulés par des ASICs. La plasticité était calculée par un ordinateur.

¹Python Neural Network

Nous allons maintenant utiliser un nouveau système neuromorphique dénommé Gaillimh [77]. La partie matérielle de ce système consiste en une

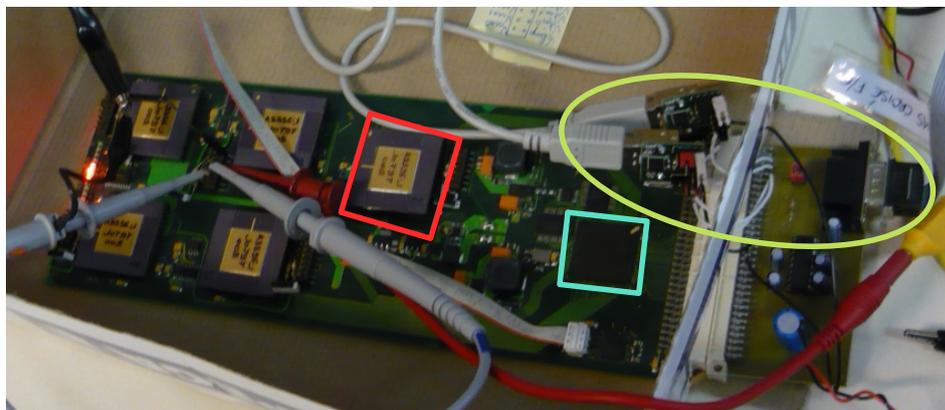


FIG. 3.1 – Photographie du système Gaillimh. Sur la gauche nous avons les ASICs Galway dont un est entouré d’un carré rose. Le FPGA se trouve à la droite des ASICs, entouré d’un carré vert. Et plus à droite sont entourés par une ellipse jaune les 3 ports série de communication avec le PC.

carte PCB, développée par Colin Lopez et Olivia Malot, ingénieurs d’étude de l’équipe ISN, contenant des ASICs pilotés par un circuit numérique FPGA (figure 3.1). Le système est relié à un ordinateur par 3 ports série RS232. Les ASICs, à l’instar du système PAX, simulent l’activité électrique des neurones basés sur le formalisme de Hodgkin et Huxley [27]. Le FPGA permet de connecter les neurones contenus dans les ASICs en réseau et effectue également le calcul de STDP[78]. Les données résultant de la simulation sont transférées en temps réel à un ordinateur pour un traitement ultérieur des données.

Neurones analogiques

- Description des composantes des circuits

Les ASICs développés par Yannick Bornat [79], nommés Galway, fournissent une grande flexibilité de configuration. Chaque circuit permet de simuler 5 neurones différents. Plusieurs conductances sont disponibles par neurone (figure 3.2). Dans tous les neurones (numérotés de 0 à 4), nous retrouvons les conductances des canaux sodique (Na), potassique (K), de fuite, ainsi que 3 conductances synaptiques (A, B et C). Les neurones 1 à 4 pos-

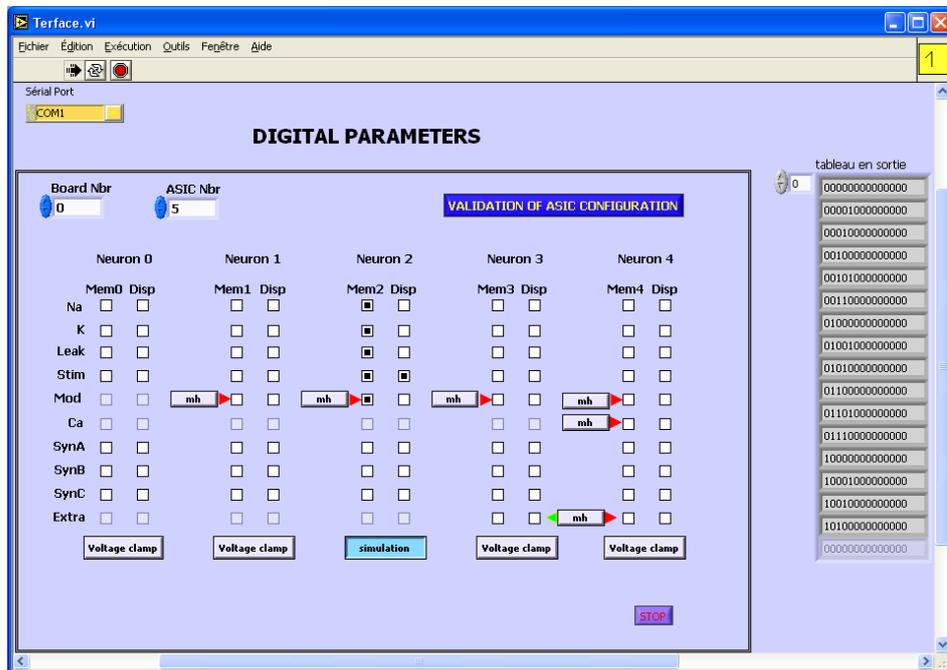


FIG. 3.2 – Façade de configuration des paramètres numériques d'un ASIC. Il est possible de choisir les conductances qui sont reliées à la membrane des neurones (Mem#) et/ou à une résistance de visualisation du signal (Disp).

sèdent en plus une conductance du canal modulant (Mod). Les neurones 3 et 4 possèdent une conductance supplémentaire (Extra). Enfin, le neurone 4 possède en plus une conductance calcique (Ca). Notez que nous n'aborderons ici ni la conductance supplémentaire ni la conductance calcique.

- Description des entrées/sorties

Afin de pouvoir exploiter le système, le FPGA a été programmé de façon à fournir un ensemble de fonctions de configuration. Ainsi il permet de régler des paramètres numériques et analogiques.

-> Les paramètres analogiques :

Les paramètres analogiques sont les valeurs des paramètres des conductances pré-citées. Ces paramètres sont stockés à l'intérieur du circuit Galway par des cellules mémoire analogiques. Précisons à titre indicatif qu'il existe, entre les grandeurs des modèles et celles utilisées par l'ASIC, un rapport 5 pour les tensions, 50 pour les courants et par conséquent 10 pour les conduc-

tances.

-> Les paramètres numériques :

Les paramètres numériques définissent la connectivité au sein de l'ASIC. En effet, pour chaque ASIC il y a la possibilité de connecter les différents canaux que l'on souhaite mettre en jeu (figure 3.3) d'une part à la membrane de chaque neurone et d'autre part à une résistance de visualisation.

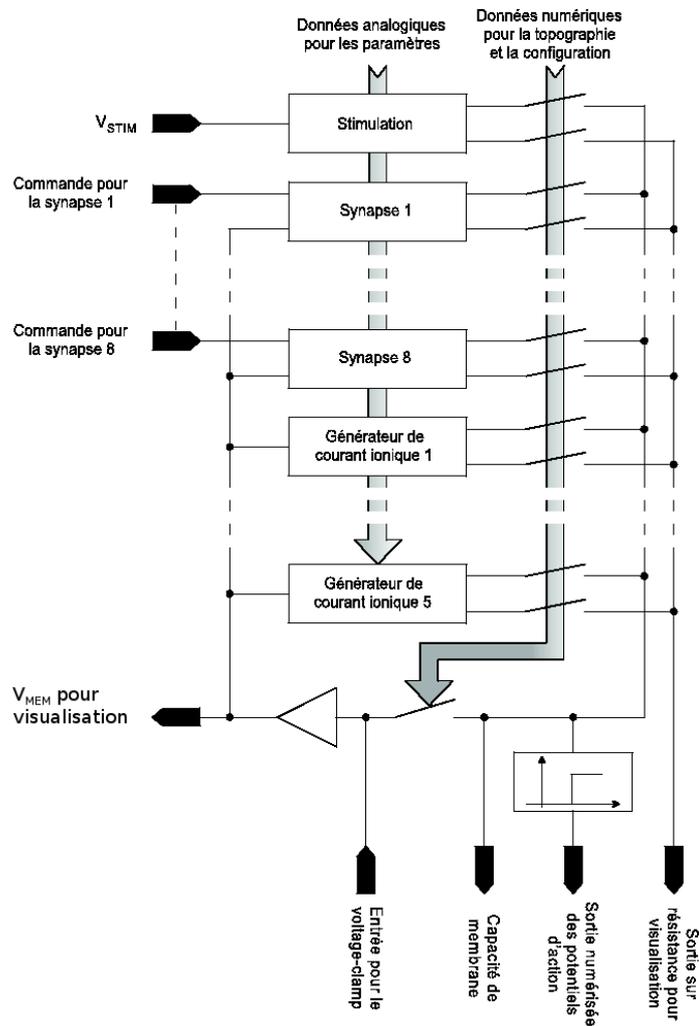


FIG. 3.3 – ASIC Galway : Connectivité contrôlée par les paramètres numériques.

Gestion du réseau et de la configuration des circuits

- De la configuration à la simulation

Avant de pouvoir utiliser le système en simulation de réseaux de neurones, il faut tout d'abord régler les ASICs afin qu'ils correspondent aux modèles de neurones désirés (RS/FS). Ainsi, des valeurs de paramètres analogiques vont être déterminées pour chaque neurone afin que ce dernier corresponde au modèle souhaité. Cette étape est décrite dans la section 3.1.2.

Puis, afin d'utiliser le système Gaillimh en mode de simulation, il faut effectuer les étapes suivantes :

Dans un premier temps, il faut charger dans les ASICs les paramètres analogiques obtenus durant la phase de réglage.

Puis, dans un deuxième temps, il faut configurer le réseau avec des paramètres numériques gérés par le FPGA. Ceux-ci déterminent la connectivité des neurones, les valeurs initiales des poids synaptiques, les synapses plastiques ou non-plastiques (c'est-à-dire soumises à la STDP ou non).

Une autre étape est nécessaire concernant le réglage du courant de stimulation imposé aux neurones et nécessite une intervention manuelle comme nous le verrons dans la section 3.2.1.

Enfin, la simulation peut être lancée. Le système nous renvoie, durant la simulation, les données résultantes que sont les événements d'émission de PAs et de calcul de STDP. Il est par ailleurs nécessaire de mettre à jour au cours de la simulation les patterns de bruit synaptique si l'on désire utiliser cette fonctionnalité comme c'était le cas avec PAX.

- Communication PC-Gaillimh

Nous présentons maintenant le protocole de communication permettant de configurer et d'utiliser le système. Comme nous l'avons vu, le système reçoit ses ordres d'un ordinateur. L'annexe D fournit plus de détails concernant ce protocole.

La figure 3.4 montre que le système communique avec un PC via trois liaisons séries *STD*, *P0* et *P1*. Voici comment sont utilisées ces trois liaisons :

-Transmission PC -> FPGA :

- >STD : envoi des paramètres analogiques/numériques
- >P0 : envoi des paramètres du réseau (connectivité)
- >P1 : envoi des informations concernant le bruit synaptique

-Transmission FPGA -> PC :

->STD : informations concernant le bruit synaptique (pour sa mise à jour)

->P0 : envoi informations PAs/poids

->P1 : envoi informations PAs/poids

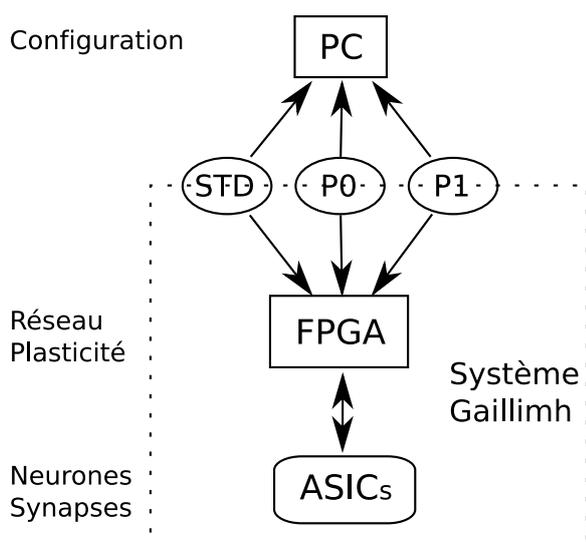


FIG. 3.4 – Communication entre le PC et le FPGA à travers trois liaisons sérieuses.

Dans le cas de la transmission des informations relatives aux PAs et aux poids du FPGA au PC, les ports P0 et P1 sont utilisés alternativement car un grand nombre de données sont à récupérer et la capacité de stockage du FPGA limitée. En effet, ces ports ont une vitesse de transmission de 321600 bits/s chacun et nous avons dimensionné le système pour 16 neurones par carte. De plus nous avons spécifié qu'un neurone ne dépasserait pas une émission de PAs à plus de 100 Hz. Par conséquent, considérant qu'une trame codant un événement de PA ou de poids fait 40 bits, dans le pire cas il serait nécessaire d'envoyer 16 PAs et 256 poids durant une période de 10 ms. Ceci implique un débit de 27,2 trames/ms or un seul port permet un débit de 23 trames/ms. Ainsi l'utilisation de deux ports permet de maintenir un débit suffisant pour assurer le transfert des données dans les conditions d'activité maximale du réseau de neurones.

Gestion du réseau et de la plasticité en hardware

Nous venons de voir les étapes permettant de configurer et d'utiliser le système. Maintenant nous allons décrire la gestion du réseau et de la plasticité par le FPGA.

- Description des entrées/sorties

Chaque neurone possède une sortie binaire indiquant si le neurone émet un PA ou non. Le FPGA stocke les informations concernant le réseau et réagit aux événements détectés au niveau de la sortie binaire que l'on vient d'évoquer. Lorsqu'un PA est émis par un neurone, le FPGA vérifie dans une table à quels neurones il est connecté, et pour chacun il récupère le poids de la synapse qu'il convertit en temps de stimulation durant lequel il met à 1 l'entrée du neurone cible correspondant à sa multi-synapse. En effet, nous l'avons évoqué au chapitre 1, le poids synaptique est codé par un créneau de tension dont la largeur est réglable et est appliqué à l'entrée synaptique du neurone. Si la synapse est plastique, alors le FPGA calcule selon l'algorithme de STDP implanté la nouvelle valeur du poids synaptique en fonction des temps d'émission des PAs des deux neurones protagonistes.

- implantation de la STDP

L'architecture complète, réalisée en VHDL par Bilel Belhadj pour sa thèse (soutenance 06/2010), dédiée au calcul de plasticité comporte plusieurs éléments principaux (figure 3.5) : une *Look-up table* (LUT), un calculateur dédié à la plasticité, une mémoire vive pour les poids synaptiques. La LUT contient les informations utiles sur la connectivité, le neurone concerné par le calcul et la liste de ses neurones afférents et efférents. Lorsqu'un PA déclenche le calcul de plasticité, le FPGA effectue le calcul et met à jour le poids synaptique recalculé [80].

Un périphérique de stockage (FIFO²) emmagasine les événements (PAs) qui seront retirés un à un déclenchant un processus de calcul de plasticité. Dès que le calcul est initié, 4 processus concurrents calculent les 4 fonctions $P(t)$, $Q(t)$, ϵ_i et ϵ_j décrites dans l'équation 1.19 modèle de STDP (figure 3.5). Pour chaque processus, un bloc de calcul d'exponentielle décroissante est instancié avec la constante de temps appropriée à la fonction concernée (τ_P , τ_Q , τ_{ϵ_i} et τ_{ϵ_j}). Chaque bloc contient une mémoire qui permet de stocker toutes les valeurs d'exponentielles de chaque synapse. En effet, le calcul s'effectue en

²First In First Out

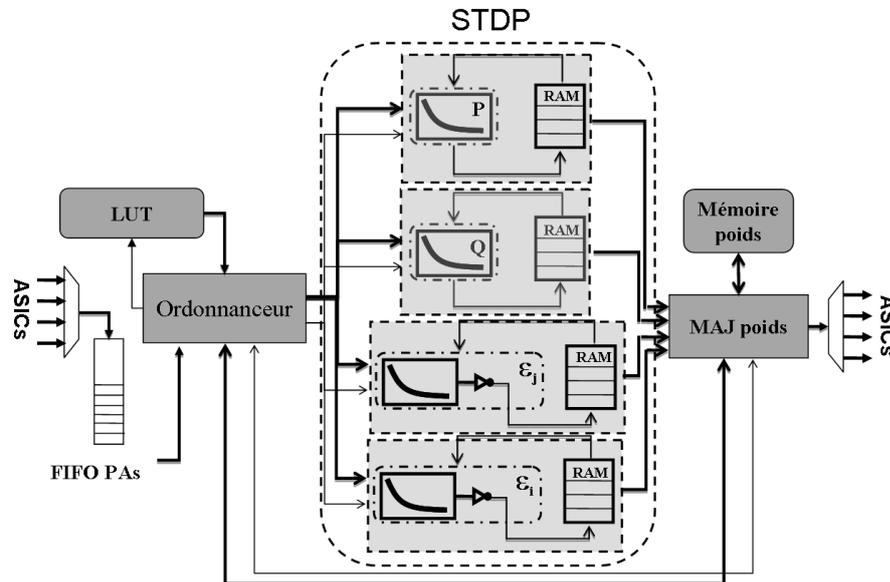


FIG. 3.5 – Architecture globale de l'implantation du calcul de STDP dans le FPGA.

continu avec un multiplexage temporel afin de respecter l'aspect continu de l'équation. Ainsi lorsque le bloc calcule pour une synapse en particulier, il récupère l'ancienne valeur de l'exponentielle lors du dernier calcul et la met à jour. Lorsque les 4 processus ont terminé leurs calculs, le calcul complet est alors effectué et le poids recalculé mis à jour.

3.1.2 Réglage des circuits - Méthodes de réglage

Afin de reproduire le modèle biophysique, un neurone possède plus de 30 paramètres ce qui offre une grande liberté de configuration, mais peut la rendre fastidieuse.

Pour un neurone de type FS, les paramètres des conductances des canaux potassique, sodique et de fuite sont à régler ainsi que les conductances synaptiques. Pour les neurones RS, la conductance supplémentaire du canal modulant est à régler en plus. Nous avons déjà évoqué ces types de neurones au chapitre 2 concernant les caractéristiques fonctionnelles des neurones HH dans la section 2.1.1.

Paramètres des conductances

Nous avons vu au chapitre 1 les caractéristiques du modèle Hodgkin-Huxley. Les équations du modèle sont rappelées en annexe A (équations A.1 à A.6) et les valeurs issues d'expérimentations en biologie sont fournies par le tableau A.1. Ce sont ces valeurs que nous utilisons comme référence pour régler les neurones. Notez que pour garantir une grande dynamique sur les paramètres et donc assurer de couvrir une large plage de valeurs, l'implantation électronique a nécessité dans certains cas l'utilisation de fonctions ayant deux paramètres d'entrée. C'est ainsi que les conductances de fuite, synaptiques et de stimulation possèdent deux paramètres électriques. Autrement dit nous avons $\bar{g}_{Fuite} = f(g_{Fuite1}, g_{Fuite2})$, $\bar{g}_{Syn} = f(g_{Syn1}, g_{Syn2})$ et $\bar{g}_{Stim} = f(g_{Stim1}, g_{Stim2})$.

Du côté électronique, les paramètres utilisés liés aux paramètres des équations sus-citées, sont les suivants :

	E	\bar{g}	V_{oac}	V_{pac}	τ_{ac}	V_{oin}	V_{pin}	τ_{in}
K	1	1	1	1	1	0	0	0
Na	1	1	1	1	1	1	1	1
Fuite	1	2	0	0	0	0	0	0
Mod	1	1	1	1	1	0	0	0
Syn	1	2	0	0	1	0	0	0
Stim	0	2	0	0	0	0	0	0

TAB. 3.1 – Nombre de paramètres électriques par conductance et par paramètre. Tous ces paramètres sont stockés sous forme de tensions dans le circuit. E est le potentiel d'équilibre, \bar{g} la conductance maximale, $V_{oac/in}$ la tension de décalage d'activation/inactivation, $V_{pac/in}$ la tension de pente d'activation/inactivation, $\tau_{ac/in}$ la constante de temps d'activation/inactivation. L'annexe A définit ces paramètres.

Les valeurs de paramètres électriques attribuées par défaut (entre 0 et 5 V) sont calculées théoriquement mais en pratique ne correspondent pas exactement aux valeurs recherchées à cause des incertitudes liées aux composants électroniques lors de la fabrication des circuits intégrés. Ainsi, avec les valeurs par défaut il est très peu probable qu'un neurone puisse émettre des PAs quelle que soit la stimulation à laquelle il est soumis. En effet, les PAs sont très sensibles aux paramètres. Il est donc nécessaire de régler pré-

cisément les paramètres que le neurone reçoit effectivement. Pour cela, nous utilisons la méthode d'extraction des paramètres associée à la technique du voltage-clamp [81] qui va nous permettre d'extraire les paramètres des différentes conductances du neurone en procédant canal par canal. Cette technique d'extraction des paramètres est utilisée en neurophysiologie et consiste, après avoir neutralisé différents canaux ioniques à part celui étudié, à imposer un potentiel de membrane puis de le porter à d'autres valeurs tout en observant la dynamique du canal ionique. Des réseaux de courbes sont alors obtenus et permettent dès lors l'extraction des paramètres recherchés.

Réglage des conductances ioniques

Les paramètres analogiques sont stockés sous forme de tensions dans les mémoires internes de l'ASIC. Les mesures effectuées sur les circuits sont traitées avant de calculer les valeurs biophysiques qu'elles représentent. Ainsi, nous nommerons « paramètres électriques » les paramètres envoyés au système et stockés par ce dernier. Le terme de « paramètres biologiques » désignera les valeurs (calculées) auxquelles ils correspondent. Dans la figure 3.6, nous pouvons voir un exemple de correspondance entre les paramètres électriques et les paramètres biologiques après mesure sur le circuit.

Afin d'essayer d'automatiser au maximum l'extraction, plusieurs programmes développés par Gilles N'Kaoua, ingénieur d'étude CNRS et membre de l'équipe ISN, avec LabView³ permettent d'effectuer les mesures et proposent de sélectionner des points sur les différents graphiques générés et les traiter pour extraire les paramètres recherchés. Les mesures, bien que facilitées sont tout de même délicates et doivent être réitérées un grand nombre de fois à cause de l'imprécision des mesures.

Nous allons maintenant voir comment extraire les paramètres biologiques à partir de mesures sur le circuit. Le principe consiste à connecter à la résistance de visualisation ainsi qu'à la membrane du neurone la conductance à étudier. Nous passons en voltage-clamp et nous appliquons différentes tensions au potentiel de membrane du neurone concerné dont on observera une variation de la réponse de la conductance connectée.

³Logiciel de développement pour systèmes d'instrumentation.

Canal Potassium

		n0	n1	n2	n3	n4	Attendu
Vp	elec	4,09	3,98	4,15	3,92	3,9	
	bio	11	11	11,4	13	11,8	11,38
Vo	elec	2,45	2,39	2,36	2,46	2,25	
	bio	-37,5	-38,8	-38	-40	-40	-37
Gm	elec	4,83	4,83	4,8	4,78	4,86	
	bio	2,25	2,5	2,3	2,4	2,2	2,2
Vequi	elec	2,1	1,87	1,96	1,87	1,98	
	bio	-100	-102	-101	-101	-110	-100
tau	elec	4,77	4,78	4,77	4,64	4,63	
	bio	2,95	3,27	3,03	3,1	3,06	3

Canal Sodium - Activation

		n0	n1	n2	n3	n4	Attendu
Vp	elec	4,15	4,19	4,35	4,24	4,26	
	bio	7,3	7,7	6,9	7,6	7,1	7,2
Vo	elec	2,38	2,39	2,33	2,38	2,3	
	bio	-38	-37,5	-35,8	-36	-38	-37
Gm	elec	3,65	3,69	3,85	3,8	3,75	
	bio	11,3	12,1	11,6	11,6	10,6	11
Vequi	elec	2,81	2,74	2,8	2,75	2,75	
	bio	49,7	50,6	53	50	51	50
tau	elec	3,2	3,2	3,4	3,3	3,45	
	bio	0,03	0,03	0,03	0,03	0,03	0,03

Canal Sodium - Inactivation

		n0	n1	n2	n3	n4	Attendu
Vp	elec	4,57	4,58	4,5	4,47	4,57	
	bio	4,1	4,6	4,8	4,6	4,2	4,6
Vo	elec	2,44	2,34	2,32	2,3	2,29	
	bio	-40	-40,8	-44	-44	-44	-42
tau	elec	4,1	4,2	4,15	4,1	4,2	
	bio	0,29	0,3	0,3	0,3	0,29	0,3

FIG. 3.6 – Réglages du Sodium et du Potassium obtenus pour l'ASIC 1, pour les 5 neurones (numérotés de n0 à n4). Les valeurs des lignes *bio* sont les paramètres biologiques obtenus après mesure du circuit configuré avec les paramètres électriques des lignes *elec*. Les valeurs des colonnes *Attendu* au bas de chaque tableau correspondent aux valeurs des paramètres biologiques du modèle de référence. V_p désigne la tension de pente, V_o la tension de décalage, G_m la conductance maximale et τ la constante de temps. L'annexe A définit ces paramètres.

Ainsi, nous appliquons différentes valeurs de tension de membrane V_m et nous récupérons en réponse une tension V_{mesure} à travers la résistance de visualisation que l'on convertit en courant I avec l'équation 3.1.

$$I = \frac{V_{mesure} - V_{gnda}}{R_{visu}} \quad (3.1)$$

où R_{visu} est la valeur de la résistance de visualisation et V_{gnda} la valeur de la masse de référence des circuits.

Avec cette méthodologie, nous pouvons régler les conductances d'un neurone afin qu'il corresponde à un modèle attendu. Pour ce faire nous modifions les paramètres électriques (cf tableau 3.1), calculons leur équivalence biologiques et les modifions à nouveau jusqu'à obtenir une correspondance la plus proche du modèle souhaité.

- Réglage de la conductance potassique :

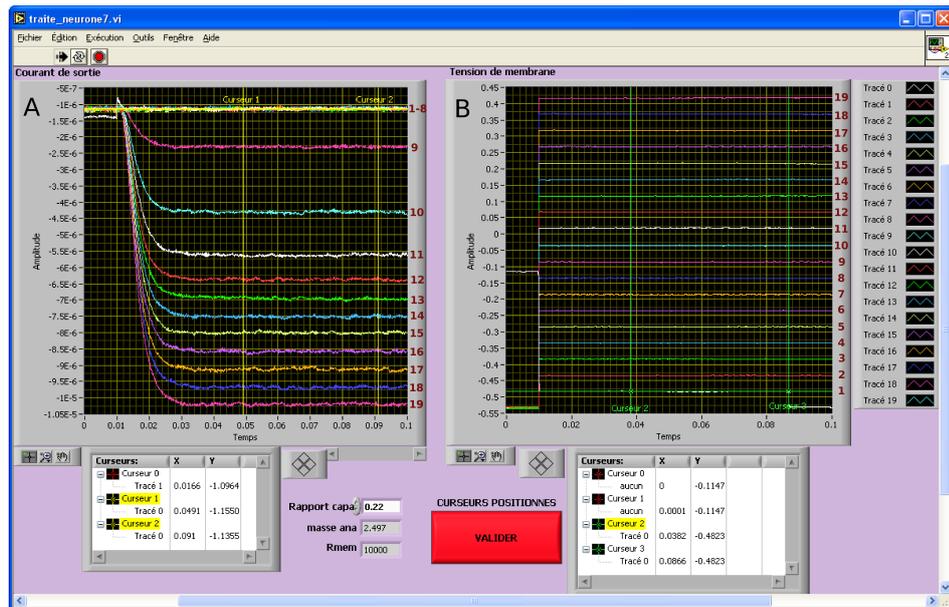


FIG. 3.7 – Mesures de la réponse du canal potassique (A) à différentes stimulations (B). (Interface LabView).

La conductance potassique possède un terme d'activation et aucun d'inactivation. Nous allons d'abord nous intéresser aux paramètres \bar{g}_K et E_K présentés au chapitre 1 et rappelés en annexe A par l'équation A.2.

Nous nous référerons aux mesures présentées à la figure 3.7 pour expliquer

le principe de l'extraction des paramètres.

La figure 3.7A présente un réseau de 19 courbes correspondant aux réponses du canal potassium à différents sauts de tension présentés sur le graphique B. Nous avons attribué le même numéro aux courbes qui se correspondent (réponse/stimulation) entre les 2 graphiques. En observant les réponses du canal, on constate que pour les réponses aux stimulations de 13 à 19, l'écart entre les tensions au régime permanent est constant. Cela signifie que l'activation est totale, autrement dit $n = 1$. En réécrivant alors le courant potassique par l'équation 3.2 il devient aisé d'isoler \bar{g}_K et E_K .

$$I_K = \bar{g}_K(V_m - E_K) \quad (3.2)$$

Nous traçons ensuite la courbe $I_K(V_m)$ comme nous pouvons le voir à la figure 3.8. La partie qui concerne l'équation sus-citée est située entre les deux curseurs. A partir de couples (V_{m_1}, I_{K_1}) et (V_{m_2}, I_{K_2}) nous déterminons \bar{g}_K et E_K par les équations 3.3 et 3.4.

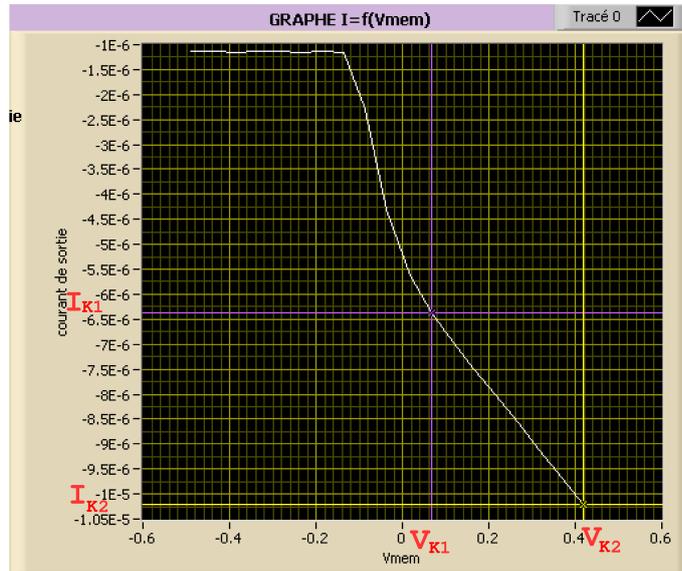


FIG. 3.8 – Courbe $I_K(V_m)$. L'activation complète concerne la partie linéaire de la courbe (entre les curseurs V_{K1} et V_{K2}).

$$\bar{g} = \frac{I_2 - I_1}{V_{m_2} - V_{m_1}} \quad (3.3)$$

$$E = V_{m1} - I_1 \cdot \frac{V_{m2} - V_{m1}}{I_2 - I_1} \quad (3.4)$$

Nous cherchons maintenant à déterminer les paramètres V_{pente} et V_{offset} de la variable d'état n (x_∞ dans l'équation A.6 de l'annexe A). Nous avons à disposition tous les éléments de l'équation A.2 ce qui nous permet d'isoler le terme d'activation dans l'équation 3.5.

$$n_\infty = \left(\frac{I}{\bar{g}_K \cdot (V_m - E_K)} \right)^{1/4} \quad (3.5)$$

Cette fois-ci nous nous intéressons à tous les points situés dans le régime permanent (courbes 1 à 19), et en traçant la courbe $n_\infty(V_m)$ (figure 3.9). La courbe obtenue est une portion de sigmoïde (entre les marqueurs jaune

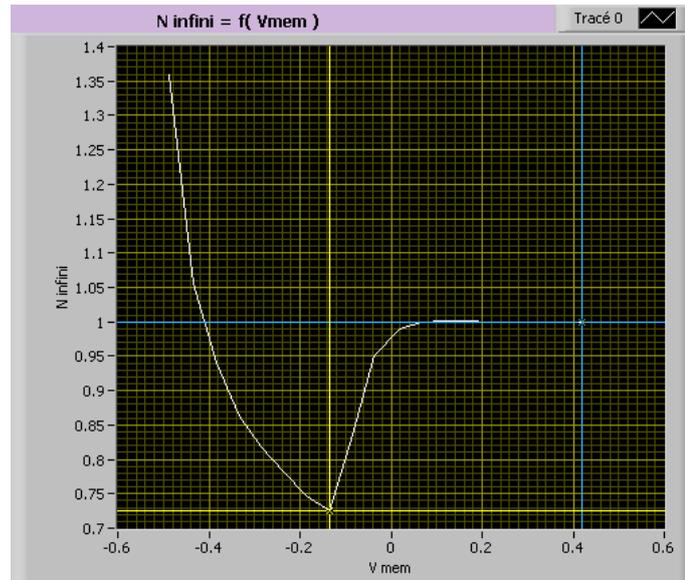


FIG. 3.9 – Courbe $n_\infty(V_m)$. Les points à considérer sont ceux compris entre les deux curseurs jaune et bleu.

et bleu) dont il faut rechercher les paramètres. Les points situés avant le marqueur jaune correspondent aux réponses 1 à 8 de la figure 3.7 qui sont inexploitable. Un programme développé par Gilles N'Kaoua extrait les valeurs des paramètres V_{pente} et V_{offset} de la sigmoïde.

Enfin, pour déterminer la constante de temps correspondant à la cinétique d'activation du potassium, nous utilisons l'expression temporelle du

terme d'activation [27] :

$$n = \left(1^{1/4} - 1^{1/4} \cdot \exp\left(\frac{-t}{\tau}\right)\right)^4 = \left(1 - \exp\left(\frac{-t}{\tau}\right)\right)^4 \quad (3.6)$$

En prenant le cas où $t = 2\tau$, nous avons $n = 0,559 \cdot n_\infty$ ce qui nous permet d'identifier pour une courbe en particulier de la figure 3.7, la constante de temps d'activation.

En utilisant la méthodologie décrite ci-dessus, nous pouvons alors connaître à partir de mesures effectuées sur les circuits la valeur des paramètres correspondant biologiquement. Ceci nous permet alors de modifier les paramètres électriques du tableau 3.1 pour obtenir les valeurs d'un modèle de neurone souhaité.

- Réglage de la conductance sodique :

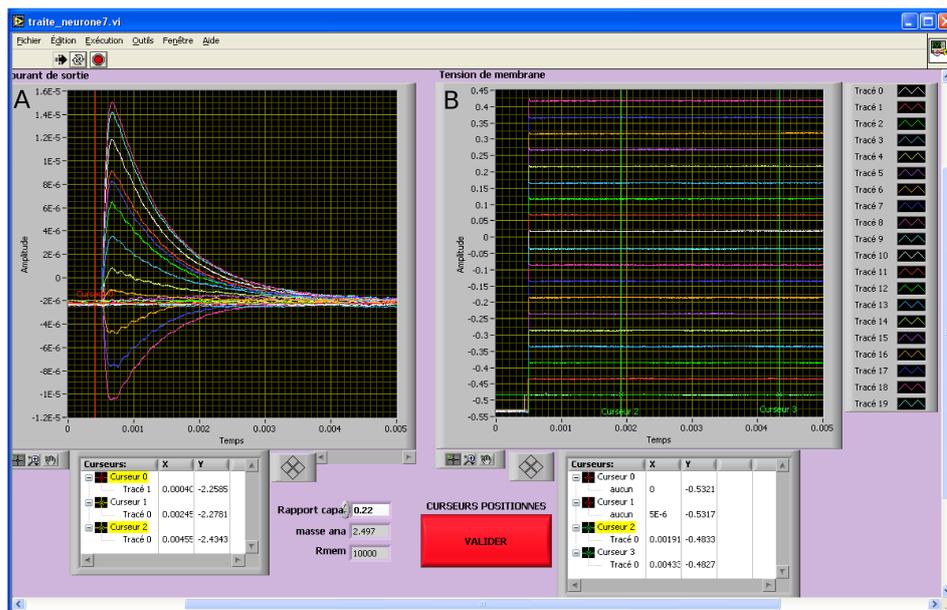


FIG. 3.10 – Mesures de la réponse du canal sodique (A) à différentes stimulations (B).

Le cas de la conductance sodique est plus complexe. En effet, nous avons à gérer un terme d'activation et un terme d'inactivation. Comme pour le potassium, nous appliquons différents sauts de tension (figure 3.10B) et nous observons les différentes réponses du canal sodique (graphique A). On observe plus particulièrement une phase de croissance qui correspond à l'ac-

tivation m du canal, et une phase décroissante plus lente correspondant à l'inactivation h .

Nous allons là aussi commencer par la recherche de \bar{g}_{Na} et E_{Na} de l'équation A.1 de l'annexe A. Compte-tenu du fait que les constantes de temps de l'activation et de l'inactivation ont une décade de différence (voir tableau A.1), nous ferons l'hypothèse que les pics de courant des réponses du canal correspondent à la réponse complète de l'activation et que l'inactivation n'a pas encore commencé. Autrement dit, les pics sont atteints pour $m = m_\infty$ (avec m_∞ calculé d'après la valeur de la tension appliquée après le saut de tension) et $h = h_\infty = 1$ (avec h_∞ calculé d'après la valeur de la tension appliquée avant le saut). La valeur de la tension avant le saut est fortement négative, ce qui implique une valeur de 1 pour le h_∞ correspondant. En résumé, les pics de courants sont atteints pour $m = m_\infty$ et $h = 1$. Par conséquent, à l'instar du potassium, nous nous trouvons pour les plus fortes valeurs de potentiels imposés après le saut dans la situation où l'activation est complète, et donc l'équation A.1 devient linéaire (équation 3.7). En traçant alors la courbe $I_{Na_{pic}}(V_m)$ (figure 3.11) nous pouvons extraire, avec la partie entre les curseurs, \bar{g}_{Na} et E_{Na} à partir de deux couples de mesure (V_{m_1}, I_{Na_1}) et (V_{m_2}, I_{Na_2}) , avec les équations 3.3 et 3.4.

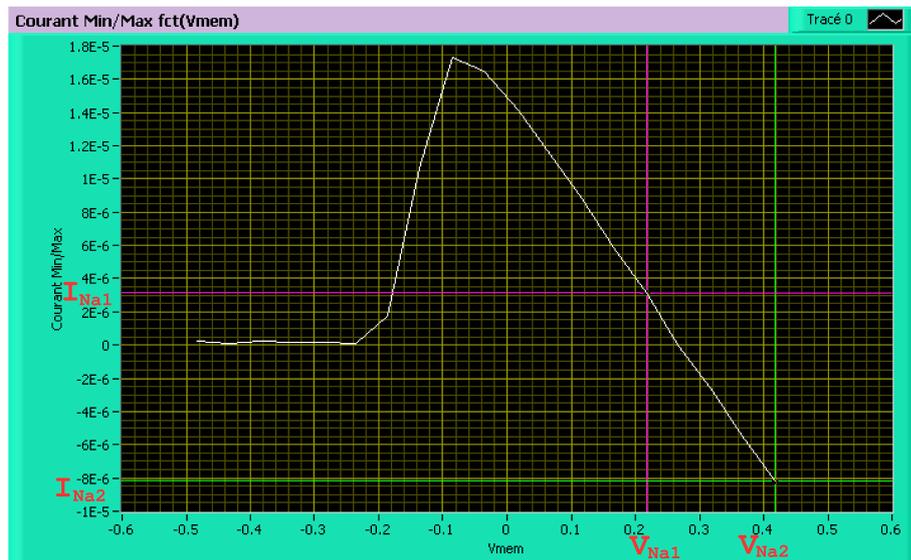


FIG. 3.11 – Courbe $I_{Na}(V_m)$. L'activation complète concerne la partie linéaire de la courbe (entre les curseurs violet et vert).

$$I_{Na_{pic}} = \bar{g}_{Na} \cdot (V_m - E_{Na}) \quad (3.7)$$

Afin de déterminer le terme d'activation, nous le réécrivons comme dans l'équation 3.8, puis nous traçons $m_\infty(V_m)$ (figure 3.12). Les points obtenus forment là aussi une portion de sigmoïde.

$$m_\infty = \left(\frac{I_{Na_{pic}}}{\bar{g}_{Na} \cdot (V_m - E_{Na})} \right)^{1/3} \quad (3.8)$$

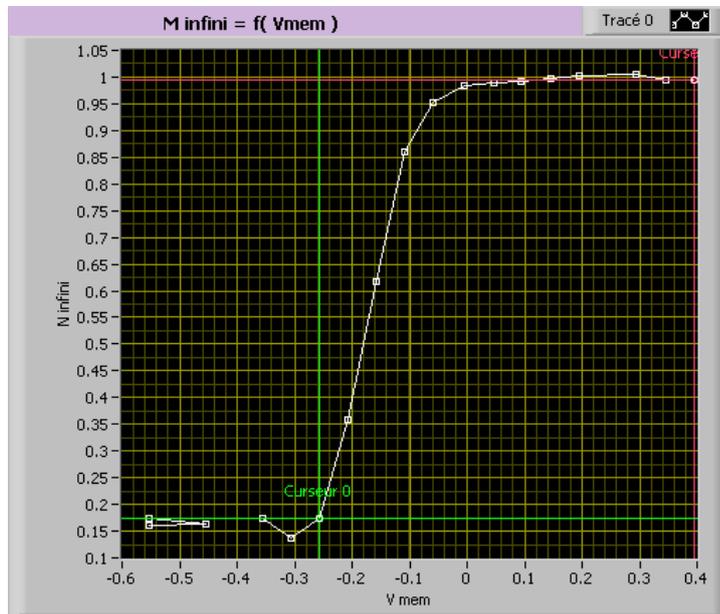


FIG. 3.12 – Courbe $m_\infty(V_m)$. Les points à considérer sont ceux compris entre les curseurs vert et rouge.

Comme pour le potassium, nous utilisons un programme qui recherche les paramètres optimaux V_{pente} et V_{offset} pour correspondre à la portion de sigmoïde qui nous intéresse.

Enfin, pour déterminer la cinétique d'activation, l'expression suivante est utilisée :

$$m = \left(1^{1/3} - 1^{1/3} \cdot \exp\left(\frac{-t}{\tau}\right) \right)^3 = \left(1 - \exp\left(\frac{-t}{\tau}\right) \right)^3 \quad (3.9)$$

Et de même que pour le potassium, à $t = 2\tau$ nous avons $m = 0,646 \cdot m_\infty$ ce

qui permet de déterminer τ à partir du graphique (A) de la figure 3.10 en considérant les points de la partie croissante de la courbe.

Maintenant nous pouvons rechercher le terme d'inactivation. Il est cependant nécessaire d'effectuer une nouvelle série de mesures (figure 3.13). Alors que pour le potassium et pour l'activation du sodium, les sauts de tension de stimulation se faisaient en partant toujours d'une même valeur pour atteindre une valeur toujours plus élevée, cette fois-ci les sauts partiront de valeurs toujours différentes pour atteindre une même valeur (graphique B). Autrement dit, l'initialisation du terme h_∞ sera différente pour chaque départ. Avec une valeur de la tension après saut toujours élevée, l'activation sera toujours complète. Cela implique que le pic de la réponse du courant correspond à l'activation complète des canaux non-inactivés au départ.

En utilisant les termes déjà connus, \bar{g}_{Na} et E_{Na} , et en prenant $m = 1$ d'après l'explication donnée précédemment pour l'activation, nous pouvons réécrire l'équation 3.10 telle que :

$$h_\infty = \frac{I_{Na_{pic}}}{\bar{g}_{Na} \cdot (V_m - E_{Na})} \quad (3.10)$$

A partir de cette courbe, nous recherchons les valeurs de paramètres d'inactivation V_{pente} et V_{offset} .

Concernant la constante de temps de l'inactivation, nous nous replaçons dans la situation où $t = \tau$ d'où $h = 0,632 \cdot h_\infty$ ce qui nous permet de déterminer τ en utilisant la figure 3.13.

- Réglage de la conductance de fuite :

Le réglage de la conductance de fuite est assez simple par rapport à ceux du potassium et du sodium. En effet, l'équation A.3 qui régit le courant de fuite ne comporte pas de terme d'activation ni d'inactivation.

Ainsi, il suffit de connecter la conductance de fuite à la membrane ainsi qu'à la résistance de visualisation, et d'appliquer à V_m différentes tensions de stimulation, puis de récupérer les réponses sur la résistance de visualisation, d'en déduire le courant correspondant à la réponse du canal en utilisant l'équation 3.1.

En traçant la courbe $I_{Fuite}(V_m)$, à partir de deux couples (V_{m_1}, I_{Fuite_1}) et (V_{m_2}, I_{Fuite_2}) il est alors possible de déduire les paramètres \bar{g}_{Fuite} et E_{Fuite} en utilisant les équations 3.3 et 3.4.

Les mesures ont été effectuées individuellement car la sensibilité du canal

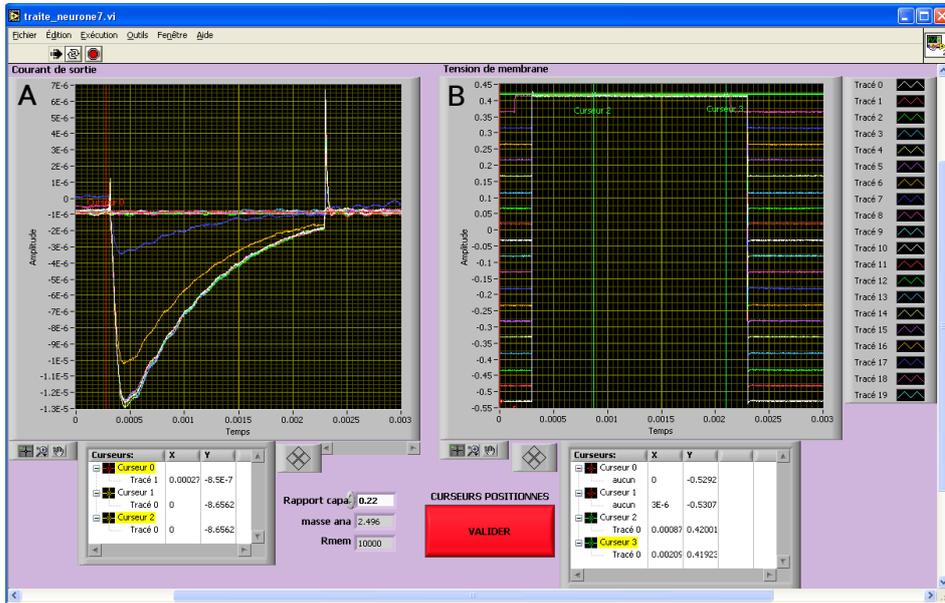


FIG. 3.13 – Mesures de la réponse du canal sodique (A) pour l’inactivation à différentes stimulations (B).

de fuite étant plus grande que les autres les mesures réalisées avec le logiciel en LabView n’étaient pas assez précises.

- Réglage de la conductance modulante :

Le courant du canal modulant, fourni par l’équation A.4, étant similaire à celui du potassium, implique une approche identique à celle utilisée pour le potassium. Pour les mêmes raisons que pour le canal de fuite, les mesures ont été effectuées en dehors de l’environnement LabView.

A la figure 3.14 nous avons en (A) les réponses aux stimulations présentées en (B). En prenant les dernières courbes du bas, nous sommes dans le cas d’une activation totale et nous pouvons alors procéder comme nous l’avons fait pour le potassium afin de déterminer \bar{g}_{Mod} et E_{Mod} .

Puis nous isolons le terme d’activation m afin de le déterminer avec un couple de points de mesure.

Et enfin, nous nous plaçons à $t = \tau$ qui implique $m = 0,632.m_{\infty}$ et nous permet de déterminer la constante de temps d’activation.

- Réglage des synapses :

Comme précédemment, on connecte la conductance synaptique à la membrane ainsi qu’à la visualisation. Nous rappelons ici l’équation du courant

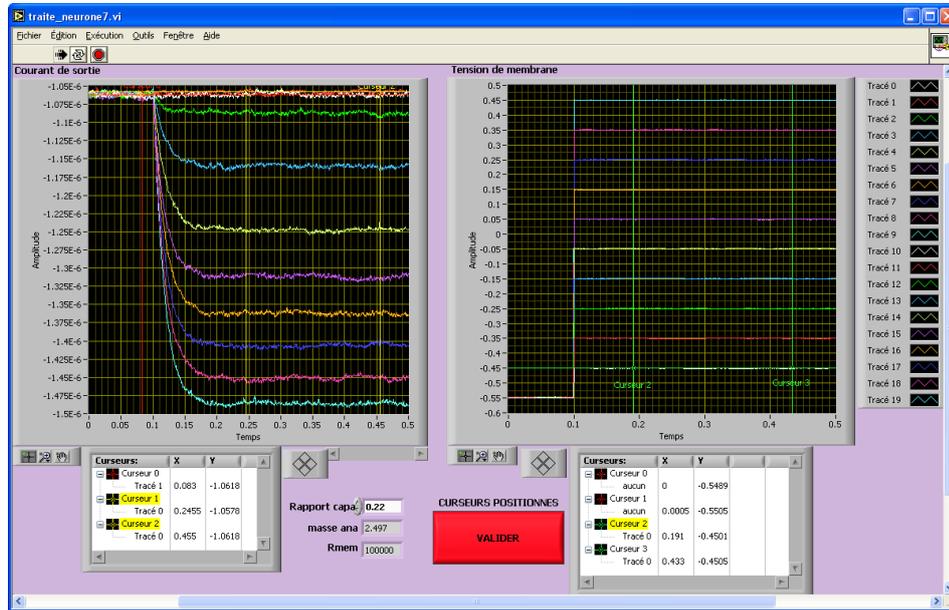


FIG. 3.14 – Mesures de la réponse du canal modulant (A) à différentes stimulations (B).

synaptique :

$$\begin{cases} I_{syn} &= g_{syn}(t) \cdot (V_m - E_{syn}) \\ g_{syn}(t) &= \bar{g}_{syn} \cdot r(t) \end{cases} \quad (3.11)$$

Le réglage de \bar{g} à une valeur importante amplifie la valeur de $(V_m - E_{syn})$. Il suffit alors de balayer plusieurs valeurs de E_{syn} avec un V_m imposé à la valeur désirée pour le E_{syn} (cf. tableau A.1) pour déterminer la valeur de E_{syn} qui annule $(V_m - E_{syn})$. Pour déterminer la conductance, une fois le paramètre électrique trouvé pour avoir le bon E_{syn} , il est alors aisé de déterminer les valeurs de conductance (il y en a 2 par synapses, cf. tableau 3.1, mais avec des valeurs identiques) en fixant $r(t)$ (avec $r(t)$ défini par l'équation 1.14 au chapitre 1) en fonction du nombre de synapses que l'on souhaite pouvoir connecter au neurone.

3.2 Utilisation du système

3.2.1 Propriétés fonctionnelles des neurones

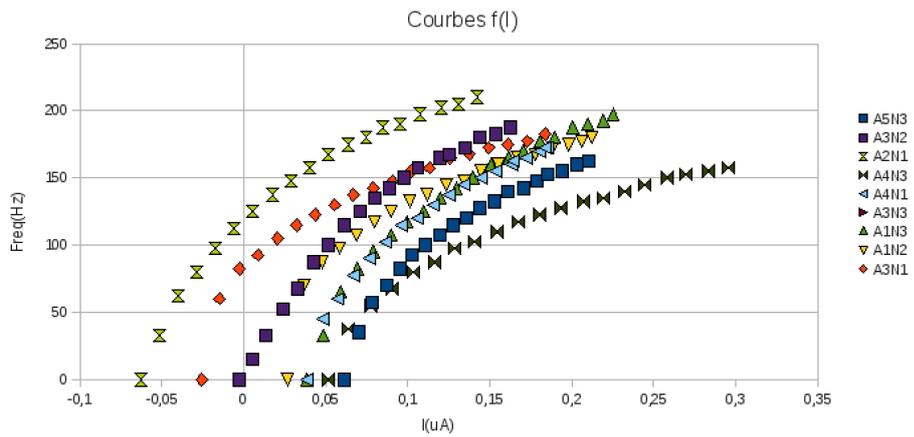
Caractérisation des neurones

Nous allons voir maintenant, une fois les neurones configurés quel est leur comportement. En effet, on s'attend à une dispersion car, les mesures effectuées précédemment pour régler les paramètres conformément au modèle, sont malheureusement soumises à la sensibilité du modèle et de ses paramètres. Ainsi, pour chaque mesure, nous n'obtenons pas nécessairement un nombre de points suffisant pour par exemple déterminer avec précision les paramètres des sigmoïdes. D'autre part, en utilisant plusieurs points de mesure, nous n'obtenons pas forcément une parfaite linéarité pour déterminer les paramètres de conductance et de potentiel d'équilibre d'une espèce ionique par exemple. Après la phase de configuration, nous avons obtenu sur 25 neurones, 13 neurones fonctionnels dont 4 FS. Notons qu'un neurone RS peut être configuré en FS en lui déconnectant le canal modulant. Les autres neurones présentent différents types de problèmes qui ne les rendent pas exploitables. Certains ont leurs synapses défaillantes (pic de décharge ou *glitches* observés lors de la stimulation de la synapse). D'autres ont une défaillance au niveau de la connexion des canaux à la membrane du neurone certainement due à un défaut lors de la fabrication. D'autres encore présentent un défaut au niveau d'un ou de plusieurs canaux les rendant inutilisables.

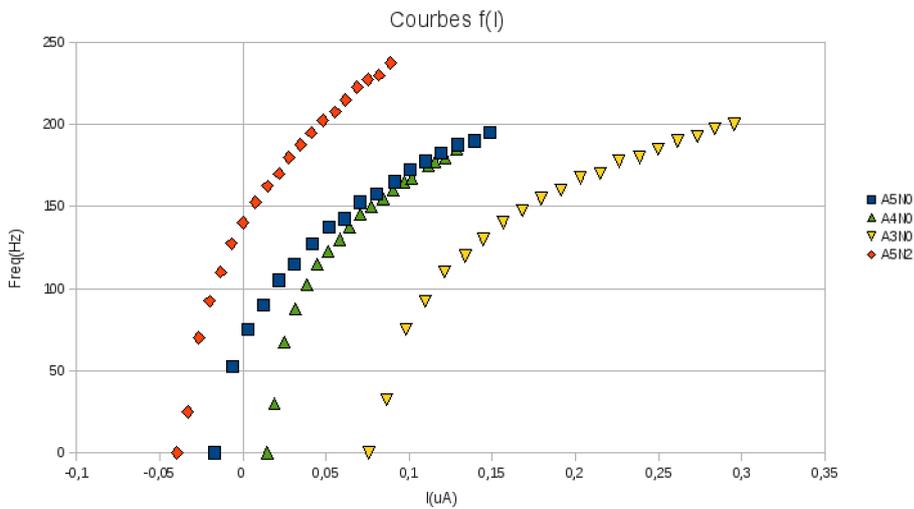
- Courbes $f(I)$:

Une fois réglé, un neurone n'émet en principe pas de PAs sans recevoir une stimulation extérieure. En imposant au neurone un courant constant suffisamment élevé, le potentiel de membrane augmentera assez afin de dépasser le seuil de déclenchement de l'émission d'un PA. Puis le potentiel de membrane est abaissé suite à l'hyperpolarisation, mais le courant constant le ramène à dépasser le seuil à nouveau et ainsi de suite. Ceci a pour effet d'imposer au neurone une certaine fréquence d'émission de PAs. Plus on augmente le courant, plus la fréquence du neurone augmente. A l'inverse, si le courant n'est pas suffisant, le potentiel de membrane reste en-dessous du seuil de déclenchement et alors le neurone n'émet aucun PA. Ainsi, nous pouvons caractériser le neurone par des courbes $f(I)$ fournissant la fréquence en fonction du courant injecté. La figure 3.15 présente les courbes $f(I)$ des 9

neurones RS et des 4 neurones FS.



(a) Neurones RS



(b) Neurones FS

FIG. 3.15 – Courbes de fréquence de décharge des neurones en fonction du courant de stimulation imposé à la membrane ($f(I)$). Dans la légende $A\#$ représente le numéro de l'ASIC et $N\#$ le numéro du neurone.

Réglage des courants de stimulation

Afin de régler les entrées de stimulation des neurones d'un ASIC, des broches sont disponibles sur la carte PCB. Cependant, aucun DAC⁴ n'est disponible sur la carte afin que la stimulation puisse être envoyée par le FPGA. Par conséquent il est nécessaire d'apporter une tension depuis une autre carte. Cette dernière a été fabriquée pour imposer une stimulation au potentiel de membrane du neurone. La tension est transformée, à l'intérieur de l'ASIC, en courant qui sera appliqué à la capacité de membrane. Des potentiomètres réglables permettent de délivrer des tensions allant de 0 à 5 V, avec une précision d'environ 2 mV. Ce courant permet de régler la sensibilité du neurone. En effet plus cette stimulation sera importante, plus le potentiel de membrane sera dépolarisé, et plus le neurone sera sensible à une stimulation synaptique. Si la stimulation est encore plus importante, elle dépolarisera suffisamment le potentiel de membrane pour que des PAs soient émis. On peut ainsi imposer une certaine fréquence d'émission de PAs au neurone.

3.2.2 Simulation avec Gaillimh

Outils développés pour exploiter le simulateur

Nous avons développé un ensemble de programmes en langage C permettant d'utiliser le système Gaillimh. Ces programmes sont compilés en un exécutable *gail_sim*. La figure 3.16 présente l'organisation globale du déroulement d'une simulation.

Les programmes en langage C sont compilés en *gail_sim*. Cet exécutable une fois lancé récupère les données de simulation à partir des fichiers indiqués dans le fichier *simu.prj*. Il configure le simulateur dans un premier temps et sauvegarde les chaînes de configuration dans les fichiers d'extension *.txt*. Puis il démarre la simulation en envoyant si nécessaire le bruit synaptique au simulateur et récupère les résultats de simulation qui sont stockés au fur et à mesure dans les fichiers *simu.spk* pour les PAs et *simu.w* pour les poids.

Le fichier *simu.it* indique les neurones (de 0 à 12) qui seront utilisés et le mode dans lequel il seront configuré (RS ou FS). Le fichier *simu.pds* contient les informations concernant la connectivité, indiquant quels neurones sont connectés, le type des synapses (plastique ou non-plastique) et les poids

⁴*Digital/Analog Converter*, convertisseur numérique/analogique.

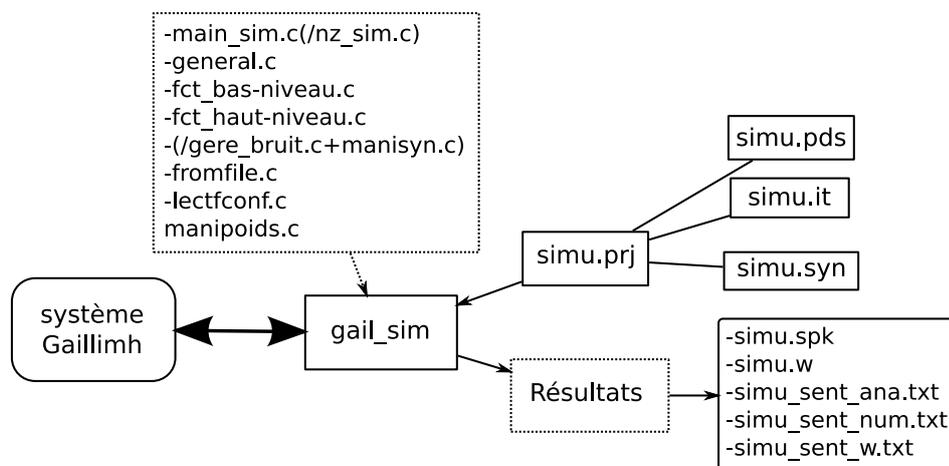


FIG. 3.16 – Organisation logicielle globale du processus de simulation.

initiaux. Le fichier *simu.syn* contient les patterns de bruit synaptique. Ces fichiers de configuration respectent le format utilisé avec le simulateur PAX.

Les programmes en C *fct_bas-niveau.c* et *fct_haut-niveau.c* fournissent les fonctions principales pour exploiter le système. Le premier programme fournit les fonctions de base permettant d'initialiser et de communiquer via les ports séries avec le FPGA. Le second, de plus haut niveau, fournit les fonctions du protocole de communication (décrit en annexe D) et envoie les paramètres lus à partir des fichiers de configuration.

Précisons que nous utilisons deux programmes *gail_sim* distincts en fonction du choix de simulation, avec ou sans bruit synaptique. Dans le cas où nous désirons utiliser le bruit, la compilation se fait en incluant les fichiers mis entre parenthèse dans la figure 3.16 (*nz_sim.c* remplaçant alors *main_sim.c*). Nous avons deux programmes distincts car l'implantation du bruit est délicate avec le protocole utilisé. Il alourdit particulièrement le code et implique une gestion de processus concurrents à cause de l'accès concurrent aux ports de communication.

Comparaison du calcul de STDP entre hardware et software

Afin de vérifier la cohérence du résultat de calcul de la STDP implantée en hardware, et présentée auparavant en section 3.1.1, nous avons réalisé une simulation avec Gaillimh (en utilisant le premier programme principal cité ci-dessus) et recalculé avec un programme développé en C sous Linux

les poids synaptiques afin de comparer les résultats. Nous avons effectué la simulation avec un réseau de 4 neurones dont la connectivité est indiquée par le schéma superposé au graphique de la figure 3.17a. Les neurones N1, N2, N3 et N4 configurés en RS sont stimulés par des courants extérieurs, imposés en utilisant la carte décrite précédemment, impliquant des fréquences d'émission de PAs initiales moyennes de respectivement 70, 50, 30 et 10 Hz. Les poids synaptiques ont été initialisés avec une valeur de 120 sachant que $\omega_{LTD} = 0$ et $\omega_{LTP} = 255$.

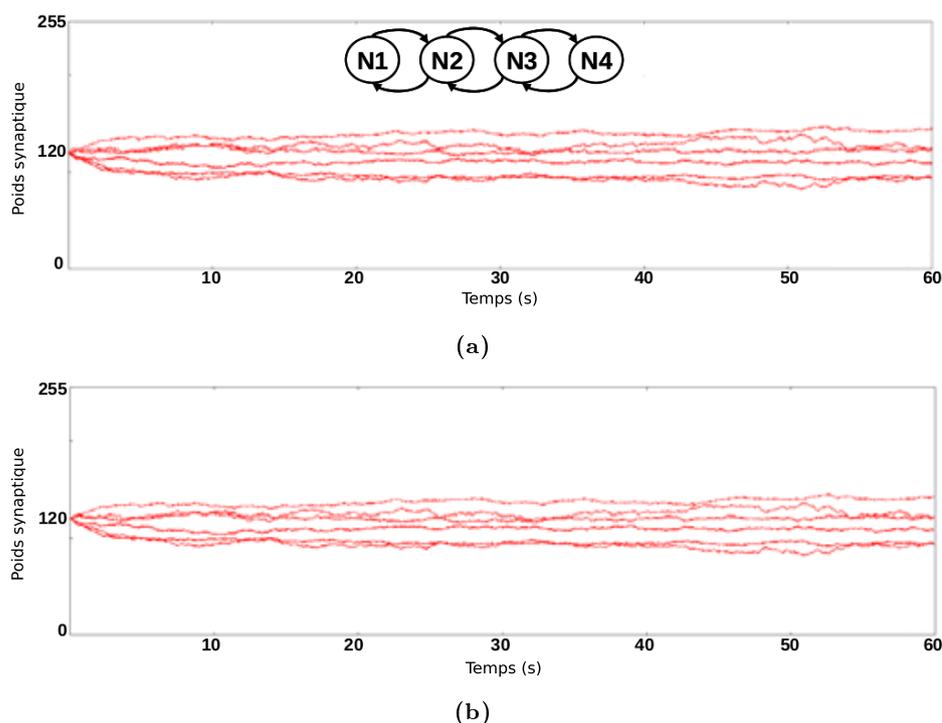


FIG. 3.17 – Comparaison de l'évolution des poids dans une simulation d'un réseau de 4 neurones avec STDP avec (a) le système électronique et calcul à virgule fixe de 16 bits d'une part et (b) un programme informatique à calcul virgule flottante de 64 bits d'autre part.

La simulation a été effectuée avec le système Gaillimh, puis le calcul de STDP a été réitéré avec le programme C à partir des données recueillies de la simulation matérielle. Les graphiques (a) et (b) de la figure 3.17 représentent pour les simulations respectives matérielle et logicielle l'évolution des poids synaptiques. On observe que les calculs logiciel et matériel donnent un résultat similaire. Précisons que la différence moyenne entre les valeurs cal-

culées par le FPGA et par le programme est de 1% relativement à la valeur maximale ce que nous considérons comme acceptable.

Problème de sensibilité au bruit

Suite à la configuration du système nous avons à disposition 13 neurones. Nous venons de voir que le système calcule correctement la plasticité synaptique avec des neurones oscillant. Cependant, l'utilisation d'une carte extérieure pour imposer les courants de stimulation implique l'existence de câbles fins qui ont un effet d'antenne qui perturbe le signal. De plus, la présence de ports série ajoutés à Gaillimh crée des interférences. Ainsi le système est particulièrement sensible au bruit ambiant généré par les différents appareils électroniques. Nous avons enveloppé notre système d'un dispositif en aluminium afin de créer une cage de Faraday. Cela n'a pas eu d'incidence sur la perturbation qui pose problème. Par conséquent, il est difficile de maintenir le système dans un état suffisamment stable afin de réaliser le même type d'études que celles menées au chapitre 2.

Nous allons illustrer les limitations que nous venons d'évoquer.

En observant à l'oscilloscope le potentiel de membrane d'un neurone en particulier qui n'émet pas de PAs, nous constatons qu'en quelques minutes, peuvent apparaître des fluctuations sur le signal qui était plat. Ces fluctuations se présentent sous forme d'ondulations basse fréquence, environ 100 Hz, dont l'amplitude peut atteindre 50 mV (pour une amplitude de PA de 600 mV).

Prenons un autre exemple illustrant le problème de perturbation de l'activité du simulateur. Nous réglons le courant de stimulation d'un neurone en particulier afin que celui-ci oscille. Dans notre exemple, nous avons choisi arbitrairement le neurone numéro 1 mais le même constat est reproductible avec tous les neurones. La tension qui est convertie en courant de stimulation imposé est de $3,460 \pm 0,01$ V, et ceci implique une oscillation du neurone.

La figure 3.18 montre pour ce même neurone, dans les mêmes conditions, sur des simulations durant 60 secondes refaites à plusieurs reprises la fréquence moyenne, la fréquence instantanée moyenne et son écart-type. On constate que l'écart entre les différentes simulations peut être important et témoigne de la sensibilité du système au bruit pour les raisons que nous avons évoqué ci-dessus.

L'inconvénient majeur est l'imprévisibilité de ces variations. Il arrive que

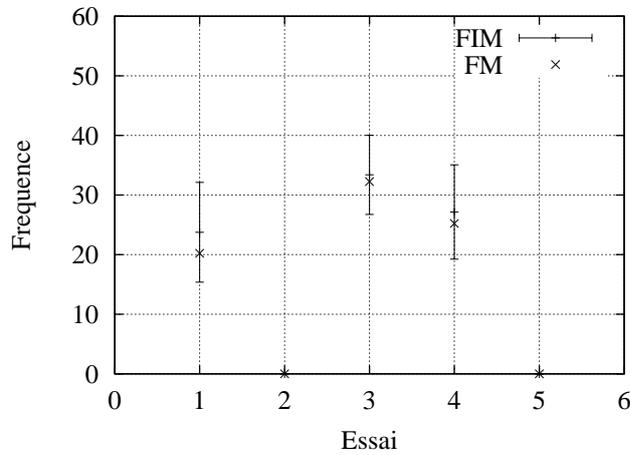


FIG. 3.18 – Fréquence moyenne (FM) et fréquence instantanée moyenne (FIM) avec écart-type d'émission de PAs du neurone numéro 1 stimulé par une tension de 3,460 V à 4 reprises pendant 60 secondes. Les conditions sont les mêmes dans les 4 essais.

durant une même simulation on passe plusieurs fois d'une condition où le signal est plat à perturbé. A cause de ce phénomène, nous ne pouvons pas effectuer de simulation en injectant du bruit synaptique et en automatisant le système.

Le prochain système de simulation en cours de développement intégrant les mêmes ASICs ne devrait pas souffrir de cet inconvénient. Les stimulations seront intégrées à la carte PCB grâce à des DACs qui vont transformer les valeurs de stimulations envoyées par le FPGA en tension sur les entrées de l'ASIC. De plus, le réglage des paramètres analogiques devrait pouvoir se faire de façon automatisée, ce qui est en partie le sujet de thèse de Laure Buhry[82]. Cet objectif peut être maintenu puisqu'il sera possible de le transposer au système qui est en cours d'élaboration dans l'équipe. L'étape suivante à la simulation est la mise à disposition du système via Internet à la communauté du projet FACETS dans un premier temps, puis plus largement à la communauté scientifique de notre domaine de recherche.

3.3 Mise à disposition du système

3.3.1 Développement avec PyNN

Description de PyNN

PyNN est un package Python pour décrire des simulations de modèles de réseaux de neurones indépendamment des spécifications du simulateur [83]. Ainsi, des modules existent pour utiliser avec PyNN le simulateur NEURON (`pynn.neuron`), NEST (`pynn.nest`) ou encore le système neuromorphique de l'équipe de Karlheinz Meier (`pynn.facetshardware1`). PyNN permet donc de faire fonctionner sur différentes plateformes aussi bien hardware que software un même programme de simulation. Nous proposons ici un nouveau module PyNN pour exploiter notre système neuromorphique : `pynn.hhhw`.

Implantation du module `hhhw` pour PyNN

Afin de pouvoir utiliser le système Gaillimh nous allons développer un module Python qui respecte la syntaxe de PyNN. Il est également nécessaire de développer des fonctions permettant d'interfacer notre module avec le système Gaillimh (figure 3.19). Cette interface est nommée *PyGail* et fait le lien avec le programme *gail_sim* (figure 3.16).

Le programme écrit en PyNN décrit la simulation qui sera traduite par *PyGail* en fichiers de configuration adaptés au programme *gail_simu*. Puis l'interpréteur lancera *gail_simu* et récupèrera les résultats pour les charger en fonction de ce qui a été décidé par PyNN.

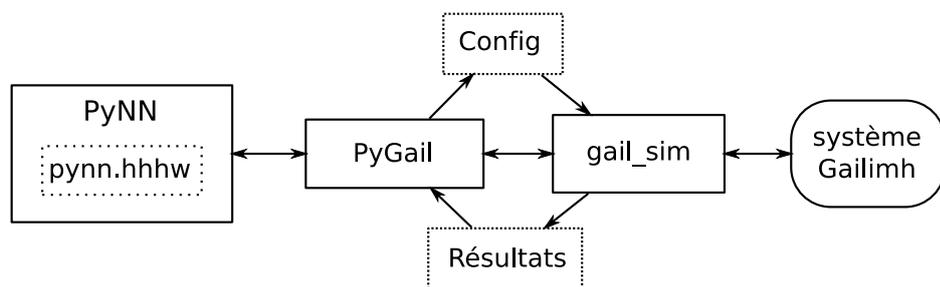


FIG. 3.19 – Principe de fonctionnement d'une simulation avec PyNN.

Nous avons re-développé un ensemble de fonctions pour notre système basées sur la documentation de PyNN API version 0,4 [84]. La liste des fonctions développées est présentée dans le tableau 3.2. Nous n'avons cependant pas implémenté les fonctions des *Population* et *Projection* qui permettent la description de simulation de gros réseaux de neurones.

Nous avons par ailleurs fourni des fonctions supplémentaires qui permettent d'exploiter directement le format de fichiers de configuration que nous utilisons avec Gaillimh.

Fonction	Description
setup()	Initialisation. Première fonction à appeler
end()	Fin de la simulation.
create()	Création de neurone(s)
connect()	Connexion des neurones ou de liste de neurones par des synapses (poids en μS)
record()	Enregistre les PAs d'un ou de plusieurs neurones dans un fichier
record_gsyn()	Enregistre l'évolution des conductances synaptiques d'une ou plusieurs synapses dans un fichier
run()	Lance la simulation (durée en <i>ms</i>)
set()	Fixe un ou plusieurs paramètre(s) d'un neurone ou d'une liste de neurones
*distance(), *distances(), *get_current_time(), *get_max_delay(), *get_min_delay(), *get_time_step(), *num_processes(), *rank(), *record_v().	

TAB. 3.2 – Liste des fonctions redéfinies pour le module *pynn.hhhw*. Les fonctions commençant par '*' ne font rien car incompatibles avec notre système, mais sont présentes pour des raisons de compatibilité avec PyNN. Ces fonctions devront être complétées pour un futur système s'il permet les opérations actuellement non fonctionnelles.

- Détail des fonctions

Fonctions **setup(simu_name="")** et **end()** :

setup procède aux initialisations éventuelles et prépare les dossiers qui recevront les résultats. *end* veille à terminer correctement. *simu_name* est le nom de base qui sera utilisé pour les noms de fichiers des résultats et pour le dossier qui les contiendra. Par défaut les résultats sont dans le dossier courant et les noms de fichiers dépendent de l'utilisation de la fonction *record()*.

Fonction **create(cellclass, param_dict=None, n=1)** :

Cette fonction permet de créer un ou plusieurs neurones de même type. La fonction retourne un identifiant (si $n=1$) ou une liste (si $n>1$) d'identifiants de neurone(s). Nous rappelons que nous disposons d'un maximum de 13 neurones actuellement et que 9 peuvent être au choix excitateurs RS ou inhibiteurs FS, les autres uniquement FS. Si un nombre supérieur à 13 est indiqué, la valeur 13 sera prise par défaut.

Le paramètre *cellclass* indique le type de neurone qui sera créé. Le type de neurone standard reconnu par tous les simulateurs est identifié par l'argument *IF_curr_alpha*. Cette fonction doit normalement offrir également la possibilité de définir les paramètres du modèle de neurone choisi. Le réglage des neurones de Gaillimh ayant été réalisé avec la méthode décrite au début de ce chapitre, il n'est pas envisageable de donner à l'utilisateur la possibilité de fournir son propre modèle⁵. Pour le moment, nous mettons à disposition l'une des 2 chaînes suivantes : "hh_rs" et "hh_fs" (qui ne font pas partie des standard de PyNN). Notez que nous associons *IF_curr_alpha* au type "hh_rs".

Fonction **connect**(*source*, *target*, *weight=None*, *delay=None*, *synapse_type=None*, *p=1*, *rng=None*) :

Cette fonction permet de construire la connectivité en créant des synapses entre les neurones. *source* et *target* peuvent normalement accepter des listes de neurones dont la probabilité est gérée par *p* mais nous conseillons de connecter les synapses neurone par neurone. *weight* fournit le poids synaptique en μS . *delay* et *synapse_type* ne sont pas utilisés car on ne gère pas de délai synaptique et le type de synapse est déterminé par le type de neurone présynaptique. La synapse sera excitatrice pour un neurone RS, et inhibitrice pour un neurone FS. Étant donné que nous utilisons la fonction avec la connexion de cellules uniques, la probabilité de connexion est $p=1$ par conséquent nous n'utilisons pas de générateur de nombre aléatoire, et *rng* est ignoré.

Fonction **record**(*source*, *filename*) :

Cette fonction indique s'il faut enregistrer les PAs d'un ou de plusieurs neurones (dépendant de si l'argument *source* est un identifiant de neurone ou

⁵Cependant, lorsque la configuration automatique sera possible avec les travaux de L. Buhry, on pourra enrichir cette fonction pour accepter les modèles personnalisés.

une liste d'identifiants de neurones), et le nom du fichier dans le paramètre *filename*.

Fonction **record_gsyn**(*source*, *filename*) :

Cette fonction indique s'il faut enregistrer l'évolution des conductances synaptiques (poids) d'une ou de plusieurs synapses soumises à STDP (dépendant de si l'argument *source* est un identifiant de neurone ou une liste d'identifiants de neurones), et le nom du fichier dans le paramètre *filename*.

Fonction **run**(*simtime*, *withSTDP=True*) :

Lance la simulation pour *simtime* milliseconde. Le programme *gail_sim* reçoit les temps en secondes (en valeur entière), il faut donc veiller à ce que *simtime* soit multiple de la seconde.

Avant de démarrer réellement la simulation, la fonction *run()* vérifie qu'un réseau de neurones a bien été créé. Puis à partir de ces informations, elle génère les fichiers de configuration formatés pour *gail_sim*. Ensuite elle lance la simulation. Une fois la simulation terminée, *run()* enregistre les données demandées par *record()* et *record_gsyn()*.

Exemple d'utilisation

Nous allons maintenant présenter un exemple de programme PyNN qui utilise les fonctions que l'on vient de voir pour lancer une simulation et récupérer les résultats :

```
import PyNN.hhhw as pynn
pynn.setup()
lst_nrn=pynn.create("hh_rs",n=5)
lst_nrn.append(pynn.create("hh_fs"))
pynn.connect(lst_nrn,lst_nrn,weight=0.01)
pynn.record(lst_nrn,"resultats_PA.txt")
pynn.record_gsyn(lst_nrn,"resultats_W.txt")
pynn.run(10000)
```

Dans cette simulation, nous avons créé 5 neurones RS et un neurone FS. Nous avons connecté tous les neurones en « *all-to-all* ». Les poids synap-

tiques sont initialisés à $0,01\mu S$. La simulation d'une durée de 10 secondes est ensuite lancée. A la fin de la simulation, dans le dossier courant les fichiers "resultats_PA.txt" et "resultats_W.txt" ont été créés.

3.3.2 Accès au système

Installation d'un serveur sur la machine hôte

Afin de rendre accessible le système depuis Internet, il est nécessaire d'installer un serveur web sur la machine hôte. Le système d'exploitation utilisé sur cette machine est une distribution linux Ubuntu Dapper Drake. Nous avons installé Apache2 auquel nous avons ajouté les modules nécessaires pour l'exécution de scripts notamment en Python.

Le serveur doit nous permettre avant toute chose de charger un fichier PyNN, de l'exécuter, et de récupérer les résultats de la simulation. L'interface pourrait également offrir la possibilité d'être dynamique, de traiter plusieurs programmes, paramétrer le traitement des résultats, etc. Nous avons pour l'instant développé l'interface minimale pour pouvoir effectuer les opérations essentielles.

L'interface web consiste en une page HTML statique avec des éléments simples. Un premier champ texte permet d'entrer un nom qui sera utilisé pour nommer les fichiers résultats. Un autre champ texte permet d'entrer une URL indiquant le fichier PyNN. Un bouton « Choose.. » permet à l'utilisateur de récupérer le fichier PyNN sur son ordinateur. Un bouton « Load » permet alors d'envoyer le fichier sur le serveur. A ce moment là, une deuxième page s'affiche et nous confirme que le programme a bien été chargé. Un bouton « Run » permet alors de lancer la simulation. Une fois la simulation terminée, un lien apparaît et mène vers le dossier contenant les résultats dans une archive qu'il suffit de récupérer par un clic droit et *Enregistrez la cible du lien sous...*

Utilisation de pynn.hhhw avec le serveur

La figure 3.20 montre l'organisation des dossiers et des programmes principaux.

Le dossier www/ est le dossier racine de notre serveur. Lorsque l'utilisateur charge le programme PyNN sur le serveur (bouton *Load*), un dossier est créé dans www/pynn/config/ avec le nom indiqué dans le champ texte dont

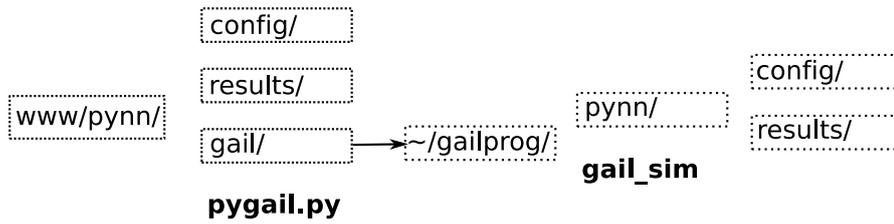


FIG. 3.20 – Organisation des dossiers et des programmes *pygail.py* et *gail_sim*. La flèche indique qu'il s'agit d'un lien.

nous avons déjà parlé. Lorsque la simulation est lancée par l'appui sur le bouton *Run*, le programme *pygail.py* est exécuté. Il récupère le programme PyNN du dossier de *config/* et le place dans le dossier *gail/* qui est un lien vers le dossier du PC contenant les programmes nécessaires au simulateur. Puis il exécute le programme PyNN qui va alors procéder à la simulation en faisant lui-même appel au programme *gail_sim*. Ce dernier placera les résultats dans le dossier *gail/pynn/results/*. Après que la simulation a pris fin, *gail_sim* rend la main au programme PyNN qui enregistre les données qui l'intéresse dans le même dossier puis rend la main à *pygail.py*. Ce dernier récupère alors les résultats et les empaquette dans une archive qu'il place alors dans le dossier utilisant le nom décidé par l'utilisateur dans le dossier *www/pynn/config/*. La page web sur laquelle attendait l'utilisateur indique alors que la simulation est terminée et indique un lien vers l'archive contenant les résultats.

3.3.3 Conclusion

Nous avons vu dans ce chapitre comment prendre en main le nouveau simulateur neuromorphique développé par l'équipe ISN. Nous avons vu que Gaillimh offre une bonne flexibilité de réglage, mais est néanmoins très délicat à configurer. Nous avons alors utilisé la technique du voltage-clamp pour régler les circuits selon les modèles désirés. Nous avons également vu que la sensibilité de ce système nous empêche de poursuivre temporairement les travaux entamés au chapitre 2. Nous avons ensuite développé une plateforme permettant d'effectuer des simulations facilement avec le langage Python et plus particulièrement PyNN qui décrit des simulations indépendamment du simulateur. Nous avons enfin mis en place le serveur permettant d'effectuer

des simulations depuis Internet. Nous allons maintenant reprendre d'un point de vue d'ensemble le travail réalisé dans cette thèse et voir ce qu'il a pu apporter à l'étude de la plasticité avec un système neuromorphique. Nous verrons également comment il se situe par rapport à la poursuite de développement et d'utilisation de systèmes neuromorphiques et la mise à disposition à la communauté scientifique des simulateurs de l'équipe ISN.

Discussion et conclusion

Avant de conclure, nous synthétiserons les travaux réalisés durant cette thèse. Nous présenterons ce qu'ils apportent, aussi bien au sein de l'équipe ISN avec le développement de simulateurs électroniques, qu'au sein de FACETS et de la communauté neuroscientifique avec la mise à disposition d'un nouvel outil d'exploration.

Effet de la corrélation du bruit sur la plasticité

Contexte général

La compréhension du fonctionnement des systèmes nerveux passe par l'étude pluridisciplinaire (neurosciences, psychologie, mathématiques, informatique, électronique, ...) d'un grand nombre de fonctions et de mécanismes. La convergence de domaines différents enrichit les possibilités d'exploration. Nous l'avons dit, la plasticité constitue une des bases fondamentales de l'exceptionnelle capacité des systèmes nerveux à manifester des phénomènes d'apprentissage, de mémoire, d'adaptation. Ses répercussions sont observables à différentes échelles, dans différents contextes. Des outils comme les IRMf⁶ ont permis de préciser nos connaissances sur l'activation simultanée de régions du cerveau impliquées dans une même tâche cognitive. La neuroanatomie et la neurophysiologie ont permis de voir la proximité de cellules nerveuses, le partage de connexions, l'existence de champs récepteurs par une organisation topologique et une proximité géographique. Les différentes techniques utilisées dans les neurosciences permettent également de préciser nos connaissances sur la plasticité synaptique. La simulation permet maintenant de représenter une multitude de cellules nerveuses, leur morphologie, leur fonctionnalisme à différentes échelles de précision, leur plasticité.

⁶Imagerie à Résonance Magnétique fonctionnelle.

Et dans cet éventail de possibilités, nous avons focalisé notre attention à l'échelle cellulaire sur la façon dont la plasticité réagit au niveau de synapses reliant des neurones recevant des informations partagées au niveau de leurs entrées.

Travaux réalisés

Nous nous sommes intéressés au phénomène de plasticité synaptique qui est connue pour être à la base des mécanismes d'apprentissage et de mémorisation. Plus particulièrement, nous avons voulu étudier l'influence que peut avoir le bruit sur l'évolution de cette plasticité au sein d'un petit réseau de neurones. Il est cependant nécessaire de préciser ce que l'on entend ici par « bruit ». Nous considérons par ce terme le fait qu'un neurone reçoit une multitude d'afférences lui conférant un comportement propre. Des neurones proches partagent un certain nombre d'afférences⁷ ce qui implique une certaine corrélation dans les informations reçues. Ainsi l'objet de l'étude a été d'examiner l'influence de la corrélation du bruit d'un petit réseau de neurones.

Afin de réaliser cette étude, nous avons utilisé un simulateur électronique dont le fonctionnement temporel est basé sur le temps biologique réel. Ce simulateur, nommé PAX et développé par l'équipe ISN, est doté de six neurones excitateurs de type RS utilisant un modèle de conductances basé sur le formalisme de Hodgkin et Huxley. La plasticité, basée sur un modèle biophysique de STDP, est calculée par un programme en langage C sur l'ordinateur relié au simulateur. Le calcul et la mise à jour des poids se fait en temps réel biologique.

Deux dispositions ont été abordées. Dans un premier temps, un réseau de 6 neurones inter-connectés avaient leurs entrées corrélées. Dans un second temps, les neurones étaient différenciés par leurs entrées en deux groupes possédant chacun une même corrélation du bruit.

La première étude a montré que l'évolution des poids synaptiques due à la STDP impliquait une répartition globale unimodale des valeurs de ces poids dans une plage de valeurs confinée lorsque la corrélation du bruit était faible. Lorsque la corrélation était très forte, la répartition, bimodale, s'établissait autour des valeurs extrêmes, et pour des corrélations intermédiaires mêlait

⁷Par le fait qu'un neurone envoie ses projections à plusieurs neurones postsynaptiques voisins

une répartition unimodale et bimodale.

La seconde étude a montré une évolution qui dépendait de la correspondance des poids à des groupes de synapses. En effet, en distinguant deux groupes de neurones, les synapses peuvent appartenir à des neurones à l'intérieur d'un groupe ou de l'autre, ou encore projeter d'un groupe à l'autre et inversement. Ainsi, en considérant ces quatre catégories de synapses et la différence entre la corrélation d'un groupe et de l'autre, nous avons observé des répartitions de poids spécifiques. Nous avons également observé que la différence de corrélation entre les deux groupes influence le mode de répartition des valeurs de poids synaptiques des synapses reliant les deux groupes.

Nous avons réalisé des observations essentiellement qualitatives. Nous avons par exemple évoqué le fait que les propriétés des neurones pouvaient influencer les résultats obtenus bien que cela ne corresponde pas au cas général. Il serait par conséquent intéressant de pouvoir quantifier ce genre de phénomènes.

Les résultats que nous avons obtenus indiquent que le partage d'informations en entrée des neurones influence la façon dont ces derniers communiquent entre-eux à travers la plasticité.

Perspectives

Les résultats obtenus sont intéressants et encourageants. Cependant, il est nécessaire de poursuivre cette étude sur des réseaux plus larges afin de valider la généralisation de ces observations. L'inclusion de synapses inhibitrices pourrait apporter des informations intéressantes sur le rôle de ces neurones qui permettent une régulation de l'activité. Le système de simulation en préparation dans l'équipe devrait permettre de réaliser ce genre d'études et ainsi de compléter les travaux de cette thèse. Il serait également intéressant d'inclure de nouveaux paramètres de simulation, par exemple l'évolution de la corrélation des patterns de PAs des neurones en fonction des poids ou des fréquences d'oscillation des neurones. Nous prévoyons dans tous les cas de poursuivre l'exploration des fonctionnalités liées à la plasticité et aux mécanismes de corrélation de l'information traitée par des neurones proches mais aussi par différentes aires cérébrales et même voir ces effets sur de larges populations les études actuelles utilisant des techniques d'imagerie cérébrale.

Dissémination des travaux

Outils développés

Pour l'utilisation des simulateurs PAX et Gaillimh nous avons développé un ensemble d'outils permettant d'exploiter ces systèmes et de traiter les données.

L'annexe C présente d'une part les formats des données de simulation brutes pour configurer des simulations et pour stocker les résultats. Et d'autre part, l'annexe fournit une description des programmes développés⁸ qui permettent de lire les fichiers de configuration, d'effectuer des simulations et d'analyser les résultats par l'intermédiaire de représentations graphiques.

Par ailleurs, nous avons présenté au chapitre 3 les outils développés pour pouvoir réaliser des simulations avec le langage de description PyNN et l'interface PyGail permettant de faire le lien entre les scripts PyNN et le simulateur Gaillimh. Nous avons également présenté le serveur fournissant une interface web utilisant PyGail pour pouvoir lancer des simulations depuis Internet.

Réutilisation et évolution au sein de l'équipe

L'ensemble des programmes développés permettent à l'équipe de simplifier l'exploitation du système en préparation intégrant les puces Galway, mais également celles des futurs systèmes de l'équipe. Il devient en effet aisé de décrire une simulation, de la lancer et de récupérer les résultats. Les programmes peuvent aussi être utilisés séparément pour visualiser rapidement des données de simulation (répartition des poids, fréquences instantanées, diagrammes de PAs, ...). Les nombreux commentaires du code et les annexes D et C, permettent une réutilisation rapide des programmes pour communiquer avec le circuit, pour le configurer mais aussi pour les futurs systèmes.

Prenons le cas du système en préparation avec les cartes Ekeru, développées par Olivia Malot. Le protocole de communication utilisé s'inspirera de celui que l'on a décrit. Les circuits analogiques étant les mêmes que ceux de Gaillimh, le programme sur le FPGA fonctionnera de manière similaire à l'actuel. Ainsi il sera possible de reprendre, en les adaptant, les codes sources

⁸en langage C, en scripts bash, Perl, Python et Gnuplot.

des programmes C de l'ensemble *gail_simu* pour configurer et utiliser le nouveau système. Les résultats qui seront générés respectant le format que l'on a défini, leur exploitation sera aisée avec les scripts existants. Actuellement, ces scripts sont déjà utilisés par B. Belhadj sur ses travaux relatifs à l'implantation de la STDP sur FPGA pour le nouveau système.

Lorsque les méthodes d'extraction automatique des paramètres analogiques des travaux de L. Buhry seront finalisés, il sera possible d'ajouter cette fonctionnalité au programme *gail_simu*, ainsi qu'enrichir la fonction *set()* pour PyNN et adapter le programme PyGail ainsi que le module *pynn.hhhw*. Cet ajout devrait être assez simple à réaliser avec les programmes existants.

Utilisation des systèmes par FACETS et la communauté scientifique

Le projet FACETS doit permettre l'émergence de nouveaux outils de simulation neuromorphique. Le langage de programmation Python est simple à apprendre et à prendre en main. Plusieurs plateformes de simulation de premier plan, tels que NEURON et NEST, fournissent une interface compatible avec PyNN (*Python Neural Network*). L'équipe allemande de Karlheinz Meier fournit un simulateur électronique permettant la simulation de larges populations de neurones avec un modèle de type *Integrate and Fire*, et le module *pynn.facetshardware1* pour PyNN. Ainsi le formalisme de PyNN tend à devenir un standard pour la simulation logicielle et matérielle de réseaux de neurones.

Nous fournissons pour notre part un simulateur temps réel, à l'échelle cellulaire, dont les neurones sont basés sur un modèle de type Hodgkin et Huxley. Nous avons déjà développé des outils permettant de réaliser des simulations avec PAX via Internet en collaboration avec l'équipe de Thierry Viéville à L'INRIA (Sophia Antipolis). Une interface web en Javascript, réalisée par Nicolas Debeissat de l'INRIA, permettait de décrire les simulations en utilisant le format XML. L'application avait été déployée sur un serveur Apache2 que nous avons installé sur l'ordinateur qui était connecté à PAX. Nous avons développé des programmes permettant d'interfacer l'application du serveur avec les programmes permettant d'effectuer des simulations. A partir du système Gaillimh, nous mettons à disposition le module *pynn.hhhw*. Gaillimh est fonctionnel mais ne constitue qu'une carte de test des circuits Galway. Le système plus performant en cours de préparation sera facilement

utilisable avec le module pour PyNN et donc constituera un nouveau simulateur pour la communauté scientifique. Ainsi, des simulations qui demanderaient d'importantes ressources informatiques avec un modèle à base de conductances, pourront être néanmoins exécutées, avec des scripts standards PyNN, par des simulateurs rapides, basés sur le temps réel.

Conclusion

La collaboration entre les disciplines qui se rencontrent autour des neurosciences permettra certainement dans l'avenir, au-delà de la simple compréhension du fonctionnement cérébral, de pouvoir pallier plus efficacement aux déficiences cognitives, aux pathologies mentales, aux neurodégénérescences, d'améliorer les traitements relatifs au cerveau. Mais aussi de développer de nouvelles formes de prothèses, des systèmes capables de mimer les systèmes nerveux et de s'y mêler voire de s'y substituer.

Les projets mêlant mathématiques, informatique et neurosciences intégratives contribuent pleinement à tendre vers ces objectifs. Dans ce tableau, l'électronique trouve une place de plus en plus importante et plus particulièrement l'électronique analogique. En effet, les avancées technologiques font que ses inconvénients deviennent moins problématiques aux vues des avantages qu'elle apporte. Le temps de développement est long, et la disparité de l'analogique implique des réglages et étalonnages comme nous l'avons vu avec Gaillimh. Cependant, le temps réel devient critique lorsque l'on désire effectuer des traitements complexes. Le parallélisme constitue également un point clé de ce mode d'implantation. Ces avantages permettent de plus d'interfacer la technologie avec le vivant. Par conséquent les travaux menés au sein de l'équipe ISN sont porteurs et amenés à se poursuivre à l'avenir.

Les travaux réalisés dans cette thèse ont contribué aux projets de l'équipe en apportant des outils variés dans ce contexte. Ainsi, nous avons développé l'environnement nécessaire à la bonne exploitation des systèmes PAX et Gaillimh. Nous avons par ailleurs exploré des propriétés de l'évolution de la plasticité dans un réseau de neurones soumis à du bruit. Dans le cadre de cette étude nous avons développé un modèle de corrélation de patterns de bruit et des outils d'analyse des résultats. Ces travaux ont donné des résultats qui pourront être confirmés sur des réseaux de neurones plus grands et plus complexes.

Enfin, les travaux de cette thèse, bien que menés dans un environnement de recherche consacré essentiellement à la microélectronique, sont les premiers de l'équipe ISN à relever pleinement des neurosciences computationnelles.

Publications de l'auteur

Revue internationale avec comité de lecture

- [A1] S. RENAUD, J. TOMAS, N. LEWIS, Y. BORNAT, A. DAOUZLI, M. RUDOLPH, A. DESTEXHE, S. SAÏGHI. *Neural Networks*, In Revision.

Conférences internationales avec comité de lecture et acte

- [B1] B. BELHADJ, J. TOMAS, O. MALOT, A. DAOUZLI, Y. BORNAT et S. RENAUD. Digital Mapping of a Realistic Spike Timing Plasticity for Real-Time Neural Simulations *In Conference on Design of Circuits and Integrated Systems, (DCIS'2009)*, accepted, 2009.
- [B2] A. DAOUZLI, S. SAÏGHI, M. RUDOLPH, A. DESTEXHE et S. RENAUD. Convergence in an Adaptive Neural Network : the Influence of Noise Inputs Correlation. *In 13th International Work-Conference on Artificial and Neural Networks, (IWANN'2009)*, 2009.
- [B3] A. DAOUZLI, S. SAÏGHI, L. BUHRY, Y. BORNAT et S. RENAUD. Weights Convergence and Spikes Correlation in an Adaptive Neural Network Implemented on VLSI. *In Procs. of the Int. Conf. on Bio-inspired Systems and Signal Processing (BIOSIGNALS)*, pages 286–291, 2008.
- [B4] S. RENAUD, J. TOMAS, Y. BORNAT, A. DAOUZLI, et S. SAÏGHI. Neuromimetic ICs with analog cores : an alternative for designing spiking neural networks. *In International Symposium on Circuits And Systems (ISCAS '2007)*, pages 3355–3358, 2007.

- [B5] O. ROCHEL, S. CHEMLA, P. KORNPORST, T. VIÉVILLE, A. DAOUZLI, SAÏGHI, C. LOPEZ, et S. RENAUD. A first step towards in silico neuronal implementation of early-vision map. In *Systems, Signal and Devices Conference (SSD'2007)*, 2007.

Conférences nationales avec comité de lecture et acte

- [C1] A. DAOUZLI, et S. SAÏGHI. Utilisation d'ASICs pour la modélisation de petits réseaux de neurones. In *Journées Nationales du Réseau Doctoral de Microélectronique (JNRDM'2008)*, Bordeaux, 2008.
- [C2] N. LEWIS, Y. BORNAT, L. ALVADO, C. LOPEZ, A. DAOUZLI, T. LEVI, J. TOMAS, S. SAÏGHI, et S. RENAUD. PAX : un outil logiciel / matériel d'investigation pour les neurosciences computationnelles. In *1ère conférence française de Neurosciences Computationnelles, Pont-à-Mousson, 23-24 octobre 2006*, pages 171–174, 2006.

Rapports d'avancement et/ou techniques

- [D1] S. DESTEXHE et al. Second progress report on the equivalence between hardware and numerical simulations *Technical report*, FACETS, 2005. Deliverable D5-2.
- [D2] S. SAÏGHI et A. DAOUZLI. Report on hardware simulations of single-cell and network benchmarks. *Technical report*, FACETS, 2007. Deliverable 17.

Bibliographie

- [1] A. Aertsen and M. Arndt. Response synchronization in the visual cortex. *Current Opinion in Neurobiology*, 3 :586–594, 1993.
- [2] W. Singer and C. M. Gray. Visual feature integration and the temporal correlation hypothesis. *Annual Review of Neuroscience*, 18 :555–586, 1995.
- [3] W. Bair. Spike timing in the mammalian visual system. *Current Opinion in Neurobiology*, 9 :447–453, 1999.
- [4] L. G. Nowak, M. H. J. Munk, J. I. Nelson, A. C. James, and J. Bullier. Structural basis of cortical synchronization. i. three types of interhemispheric coupling. *Journal of Neurophysiology*, 74 :2379–2400, 1995.
- [5] L. G. Nowak, M.H.J. Munk, A. C. James, P. Girard, and J. Bullier. Cross-correlation study of the temporal interactions between areas v1 and v2 of the macaque monkey. *Journal of Neurophysiology*, 81 :1057–1074, 1999.
- [6] S. Grün, Ma. Diesmann, F. Gammont, A. Riehle, and A. Aertsen. Detecting unitary events without discretization of time. *Journal of Neuroscience Methods*, 94 :67–79, 1999.
- [7] A. Riehle, F. Grammont, M. Diesmann, and S. Gruen. Dynamical changes and temporal precision of synchronized spiking activity in monkey motor cortex during movement preparation. *Journal of Physiology (Paris)*, 94 :569–582, 2000.
- [8] N. Kopell, G. B. Ermentrout, M. A. Whittington, and R. D. Traub. Gamma rhythms and beta rhythms have different synchronization properties. *Proceedings of the National Academy of Sciences of the United States of America*, 97(4) :1867–1872, February 2000.

- [9] S. R. y Cajàl. *La Textura del Sistema Nerviosa del Hombre y los Vertebrados*. Moya, Madrid, 1904.
- [10] D. O. Hebb. *The Organization of Behaviour*. John Wiley and Sons, 1949.
- [11] T. V. P. Bliss and G. L. Collingridge. A synaptic model of memory : Long-term potentiation in the hippocampus. *Nature*, 361 :31–39, January 1993.
- [12] D. J. Linden and J. A. Connor. Long-term synaptic depression. *Annual Review of Neuroscience*, 18 :319–357, 1995.
- [13] R. C. Malenka and R. A. Nicoll. Long-term potentiation - a decade of progress? *Science*, 285(5435) :1870–1874, September 1999.
- [14] J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. In *Proceedings of the National Academy of Sciences*, volume 79, pages 2554–2558, 1982.
- [15] K. S. Lashley. In search of engram. *Neurocomputing : foundations of research*, pages 57–63, 1988.
- [16] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5 :115–137, 1943.
- [17] F. Rosenblatt. The perceptron : a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65 :386–408, 1958.
- [18] M. Minsky and S. Papert. *Perceptron*. M.I.T. Press, Cambridge, Mass., 1969.
- [19] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *Learning internal representations by error propagation*. MIT Press, Cambridge, MA, USA, 1988.
- [20] L. F. Abbott. Learning in neural network memories. *Network*, 1 :105–122, 1990.
- [21] W. Maass and T. Natschläger. Networks of spiking neurons can emulate arbitrary hopfield nets in temporal coding. *Network : Computation in Neural Systems*, 8(4) :355–371, 1997.
- [22] W. Maass. Networks of spiking neurons : The third generation of neural network models. *Neural Networks*, 10 :1659–1671, 1997.

- [23] W. Gerstner and W. M. Kistler. *Spiking Neuron Models : Single Neurons, Populations, Plasticity*. Cambridge University Press, 2002.
- [24] E. M. Izhikevich. Simple model of spiking neurons. *Neural Networks, IEEE Transactions on*, 14(6) :1569–1572, 2003.
- [25] R. Brette, M. Rudolph, T. Carnevale, M. Hines, D. Beeman, J. M. Bower, M. Diesmann, A. Morrison, P. H. Goodman, F. C. Harris, and Others. Simulation of networks of spiking neurons : A review of tools and strategies. *Journal of Computational Neuroscience*, 23 :349–398, 2007.
- [26] M. L. Hines and N. T. Carnevale. The neuron simulation environment. *Neural Computation*, 9 :1179–1209, 1997.
- [27] A. L. Hodgkin and A. F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology*, 117 :500–544, 1952.
- [28] H. Markram. The blue brain project. *Nature Reviews Neuroscience*, 7(2) :153–160, 2008.
- [29] M. Mahowald and R. Douglas. A silicon neuron. *Nature*, 354 :515–518, 1991.
- [30] K. Shimonomura and T. Yagi. 2008 special issue : Neuromorphic vlsi vision system for real-time texture segregation. *Neural Networks*, 21(8) :1197–1204, 2008.
- [31] G. Indiveri and S. Fusi. Spike-based learning in vlsi networks of integrate-and-fire neurons. In *International Symposium on Circuits And Systems*, pages 3371–3374. IEEE, 2007.
- [32] S. Mitra, G. Indiveri, and S. Fusi. Learning to classify complex patterns using a vlsi network of spiking neurons. In John C. Platt, Daphne Koller, Yoram Singer, and Sam T. Roweis, editors, *NIPS*. MIT Press, 2007.
- [33] J. H. B. Wijekoon and P. Dudek. 2008 special issue : Compact silicon neuron circuit with spiking and bursting behaviour. *Neural Network*, 21(2-3) :524–534, 2008.
- [34] K.M. Hynna and K. Boahen. Silicon neurons that burst when primed. In *International Symposium on Circuits And Systems*, pages 3363–3366. IEEE, 2007.

- [35] J. Fieres, A. Grübl, S. Philipp, K. Meier, J. Schemmel, and F. Schürmann. A platform for parallel operation of vlsi neural networks. In *Brain Inspired Cognitive Systems*, 2004.
- [36] J. Schemmel, J. Fieres, and K. Meier. Wafer-scale integration of analog neural networks. In *IJCNN*, pages 431–438. IEEE, 2008.
- [37] J. Schemmel, D. Bruderle, K. Meier, and B. Ostendorf. Modeling synaptic plasticity within networks of highly accelerated i&f neurons. In *International Symposium on Circuits and Systems*, pages 3367–3370. IEEE, 2007.
- [38] S. Renaud, J. Tomas, Y. Bornat, A. Daouzli, and S. Saïghi. Neuro-mimetic ICs with analog cores : an alternative for simulating spiking neural networks. In *International Symposium on Circuits And Systems*, pages 3355–3358. IEEE, 2007.
- [39] D. Tritsch. *Le potentiel d'action des axones*. Chapitre 6 dans "Physiologie du neurone" de D. Tritsch, D. Chesnoy-Marchais, A. Feltz, Doin, Paris, 1998.
- [40] BW. Connors and MJ. Gutnick. Intrinsic firing patterns of diverse neocortical neurons. *Trends in Neurosciences*, 13 :99–104, 1990.
- [41] W. Rall. Distinguishing theoretical synaptic potentials computed for different soma-dendritic distributions of synaptic input. *J Neurophysiol*, 30(5) :1138–1168, September 1967.
- [42] A. Destexhe, Z. F. Mainen, and T. J. Sejnowski. An efficient method for computing synaptic conductances based on a kinetic model of receptor binding. *Neural Computation*, 6 :10–14, 1994.
- [43] A. Destexhe, Z. F. Mainen, and T. J. Sejnowski. Kinetic models of synaptic transmission. *Methods in Neuronal Modeling*, 1998.
- [44] W. W. Lytton. Optimizing synaptic conductance calculation for network simulations. *Neural computation*, 8(3) :501–509, April 1996.
- [45] G. Chechik and N. Tishby. Temporally dependent plasticity : An information theoretic account. In *NIPS*, pages 110–116, 2000.
- [46] L. F. Abbott and Sacha B. Nelson. Synaptic plasticity : taming the beast. *Natural Neuroscience*, 3 :1178–1183, 2000.
- [47] D. E. Feldman. Timing-based LTP and LTD at vertical inputs to layer II/III pyramidal cells in rat barrel cortex. *Neuron*, 27 :45–56, 2000.

- [48] P. D. Roberts and C. C. Bell. Spike timing dependent synaptic plasticity in biological systems. *Biological Cybernetics*, 87 :392–403, 2002.
- [49] A. Kepecs, M. C. W. van Rossum, S. Song, and J. Tegner. Spike timing dependent plasticity : common themes and divergent vistas. *Biological Cybernetics*, 87 :446–458, 2002.
- [50] S. Song and L. Abbott. Cortical development and remapping through spike timing-dependent plasticity. *Neuron*, 32 :339–350, 2001.
- [51] H. Markram, J. Lubke, M. Frotscher, and B. Sackmann. Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science*, 275 :213–215, 1997.
- [52] W. B. Levy and O. Steward. Temporal contiguity requirements for long-term associative potentiation/depression in the hippocampus. *Neuroscience*, 8 :791–797, 1983.
- [53] J. C. Magee and D. Johnston. A synaptically controlled, associative signal for hebbian plasticity in hippocampal neurons. *Science*, 275 :209–213, 1997.
- [54] G. Bi and M. Poo. Synaptic modifications in cultured hippocampal neurons : dependence on spike timing, synaptic strength, and postsynaptic cell type. *The Journal of Neuroscience*, 18(24) :10464–10472, 1998.
- [55] R. C. Froemke and Y. Dan. Spike-timing-dependent plasticity modification induced by natural spike trains. *Nature*, 416 :433–438, 2002.
- [56] C. C. Rumsey and L. F. Abbott. Equalization of synaptic efficacy by activity- and timing-dependent synaptic plasticity. *The Journal of Neurophysiology*, 91 :2273–2280, 2003.
- [57] R. C. Froemke, M. Poo, and Y. Dan. Spike-timing-dependent plasticity depends on dendritic location. *Nature*, 434 :221–225, 2005.
- [58] M. C. W. van Rossum, G-Q. Bi, and G. G. Turrigiano. Stable hebbian learning from spike timing-dependent plasticity. *The Journal of Neuroscience*, 20 :8812–8821, 2000.
- [59] M. C. W. van Rossum and G. G. Turrigiano. Correlation based learning from spike timing dependent plasticity. *Neurocomputing*, 38-40 :409–415, 2001.
- [60] M. Badoual, Q. Zou, A. P. Davison, M. Rudolph, T. Bal, Y. Frégnac, and A. Destexhe. Biophysical and phenomenological models of multiple

- spike interactions in spike-timing dependent plasticity. *Int. J. Neural Syst.*, 16(2) :79–98, 2006.
- [61] B.G. Cragg. Changes in visual cortex on first exposure of rats to light : Effect on synaptic dimensions. *Nature*, 215 :251–253, 1967.
- [62] J. Defelipe and I. Farinas. The pyramidal neuron of the cerebral cortex : morphological and chemical characteristics of the synaptic inputs. *Progress in neurobiology*, 39 :563–607, 1992.
- [63] J. E. Gruner, J. C. Hirsch, and C. Sotelo. Ultrastructural features of the isolated suprasylvian gyrus. *The Journal of Comparative Neurology*, 154 :1–72, 1974.
- [64] J. Szentagothai. The use of degeneration in the investigation of short neuronal connections. *Progress in Brain Research*, 14 :1–32, 1965.
- [65] E. V. Evarts. Temporal patterns of discharge of pyramidal tract neurons during sleep and waking in the monkey. *Journal of Neurophysiology*, 27 :152–171, 1964.
- [66] D. H. Hubel. Single unit activity in striate cortex of unrestrained cats. *Journal of Neurophysiology*, 147 :226–238, 1959.
- [67] W. H. Calvin and C. F. Stevens. Synaptic noise as a source of variability in the interval between action potentials. *Science*, 155 :842–844, 1967.
- [68] W. H. Calvin and C. F. Stevens. Synaptic noise and other sources of randomness in motoneuron interspike intervals. *Journal of Neurophysiology*, 31 :574–587, 1968.
- [69] M. Abeles. *Local Cortical Circuits : An Electrophysiological Study*. Springer-Verlag, 1982.
- [70] L. Alvado, J. Tomas, S. Saïghi, S. Renaud-Le Masson, T. Bal, A. Destexhe, and G. Le Masson. Hardware computation of conductance-based neuron models. *Neurocomputing*, 58-60 :109–115, 2004.
- [71] Y. Bornat, S. Renaud, J. Tomas, S. Saïghi, Q. Zou, and A. Destexhe. An analog/digital simulation system for biomimetic neural networks. In *Dynamical principles for neuroscience and intelligent biomimetic devices EPFL Latsis Symposium 2006*, pages 71–72, Lausanne Switzerland, 03 2006.
- [72] N. Lewis, Y. Bornat, L. Alvado, C. Lopez, A. Daouzli, T. Levi, J. Tomas, S. Saïghi, and S. Renaud. Pax : un outil logiciel / matériel d’investi-

- gation pour les neurosciences computationnelles. In *NeuroComp*, pages 171–174, October 2006.
- [73] Q. Zou, Y. Bornat, S. Saïghi, J. Tomas, S. Renaud, and A. Destexhe. Analog-digital simulations of full-conductance-based networks of spiking neurons with spike timing dependent plasticity. *Network : computation in neural systems*, 17 :211–233, 2006.
- [74] S. Song, K. Miller, and L. Abbott. Competitive hebbian learning through spike-timing-dependent synaptic plasticity. *Nature Neuroscience*, 3 :919–926, 2000.
- [75] R. Gutig, R. Aharonov, S. Rotter, and H. Sompolinsky. Learning input correlations through nonlinear temporally asymmetric hebbian plasticity. *Journal of Neuroscience*, 23 :3697–3714, 2003.
- [76] Q. Zou and A. Destexhe. Kinetic models of spike-timing dependent plasticity and their functional consequences in detecting correlations. *Biol. Cybern.*, 97(1) :81–97, 2007.
- [77] .Y Bornat, J. Tomas, C. Lopez, O. Malot, B. Belhadj, and S. Renaud. Design of analog/digital simulator dedicated to real-time neurocomputing. In *DCIS*, pages 391–396, 2007.
- [78] B. Belhadj, J. Tomas, O. Malot, A. Daouzli, Y. Bornat, and S. Renaud. Digital mapping of a realistic spike timing plasticity for real-time neural simulations. In *Conference on Design of Circuits and Integrated Systems DCIS'2009 (accepted)*, 2009.
- [79] Y. Bornat. *Réseaux de neurones sur silicium : une approche mixte, analogique/numérique, pour l'étude des phénomènes d'adaptation, d'apprentissage et de plasticité*. PhD thesis, Université Bordeaux 1, 2006.
- [80] B. Belhadj, J. Tomas, O. Malot, G. N'Kaoua, Y. Bornat, and S. Renaud. Fpga-based architecture for real-time synaptic plasticity computation. In *International Conference on Electronics, Circuits and Systems*, pages 93–96, 2008.
- [81] L. G. Nowak, M. H. J. Munk, J. I. Nelson, A. C. James, and J. Bullier. Dynamic electrical characteristics of the squid axon membrane. *Arch Sci Physiol*, 3 :253–258, 1949.
- [82] L. Buhry, S. Saïghi, W. Ben Salem, and S. Renaud. Adjusting neuron models in neuromimetic ics using the differential evolution algorithm. In

- Proceedings of the 4th international IEEE EMBS conference on Neural Engineering*, 2009.
- [83] A.P. Davison, D. Brüderle, J.M. Eppler, J. Kremkow, E. Müller, D.A. Pecevski, L. Perrinet, and P. Yger. Pynn : a common interface for neuronal network simulators. *Frontiers in Neuroinformatics*, 2008.
- [84] <http://neuralensemble.org/trac/PyNN>.
- [85] Q. Zou. *Computational models of spike timing dependent plasticity : synapses, neurons and circuits*. PhD thesis, Université Paris VI, 2006.

Annexe A

Paramètres du modèle Hodgkin-Huxley

Nous présentons dans cette annexe les valeurs des paramètres du modèle de neurone utilisé basé sur le formalisme de Hodgkin et Huxley.

Nous reprenons dans un premier temps les équations du modèles :

$$I_{Na} = \bar{g}_{Na} \cdot m^3(V_m) \cdot h(V_m) \cdot (V_m - E_{Na}) \quad (\text{A.1})$$

$$I_K = \bar{g}_K \cdot n^4(V_m) \cdot (V_m - E_K) \quad (\text{A.2})$$

$$I_{Fuite} = \bar{g}_{Fuite} \cdot (V_m - E_{Fuite}) \quad (\text{A.3})$$

$$I_{mod} = \bar{g}_{mod} \cdot m_{mod}(V_m) \cdot (V_m - E_{mod}) \quad (\text{A.4})$$

où V_m est le potentiel de membrane, I_x est le courant de la conductance ionique x , E_x son potentiel d'équilibre, et \bar{g}_x sa conductance maximale. Les paramètres m et h , n , m_{mod} des équations respectives A.1, A.2, A.3, A.4 sont des variables d'état dépendant de V_m dont l'expression est la suivante si l'on considère x la variable d'état :

$$\tau \cdot \frac{dx}{dt}(t) = x_\infty(V_m) - x(t) \quad (\text{A.5})$$

avec la sigmoïde

$$x_{\infty}(V_m) = \frac{1}{1 + \exp(\pm \frac{V_m - V_{offset}}{V_{pente}})} \quad (\text{A.6})$$

où x est la variable d'état, le signe retenu à l'intérieur de l'exponentielle sera - pour l'activation et + pour l'inactivation (h dans l'équation A.1). V_m est le potentiel de membrane. Les paramètres V_{offset} et V_{pente} sont respectivement le décalage et la pente de la sigmoïde.

Le tableau A.1 présente pour les équations précédentes les valeurs des paramètres utilisés pour configurer les neurones de nos systèmes neuromorphiques décrits aux chapitres 2 et 3.

	E	\bar{g}	V_{oac}	V_{pac}	τ_{ac}	V_{oin}	V_{pin}	τ_{in}
K	-100	2,2	-37	11,38	3	x	x	x
Na	50	11	-37	7,2	0,03	-42	4,6	0,3
Fuite	-70	0,022	x	x	x	x	x	x
Mod	-100	0,1	-35	11,4	x	x	x	x
Syn Exc	0	0,03	x	x	5	x	x	x
Syn Inh	-80	-0,08	x	x	30	x	x	x

TAB. A.1 – Paramètres biologiques de référence pour le réglage des neurones RS/FS. Les tensions V_x sont en mV , les τ_x en ms et \bar{g} est en μS . L'indice "ac" indique le cas d'activation, et "in" le cas d'inactivation. V_o correspond à V_{offset} et V_p à V_{pente} .

Annexe B

Algorithme itératif de STDP

Cette annexe présente l'algorithme exposé en annexe A de la thèse de qui correspond à l'intégration de l'algorithme de STDP utilisé dans cette thèse. Le modèle de STDP en question est présenté au chapitre 1 dans la section 1.3.3.

L'algorithme décrit dans cette annexe est utilisé car le modèle proposé dans [60] implique la prise en compte de tous les couples de PAs pré et postsynaptiques. La nécessité d'effectuer les calculs en temps réel nous a amené à utiliser la méthode de calcul itérative présentée en annexe A de la thèse de Quan Zou [85].

Dans le cas de la potentialisation, le calcul se fait au temps d'émission du PA postsynaptique (t_i) par le neurone i . Théoriquement ce temps dépend des couples qu'ils constitue avec tous les temps de PAs émis par le neurone j (t_j^k avec k de 0 à m). Autrement dit, cela revient à calculer :

$$\sum_{k=0}^m P[t_i - t_j^k] \epsilon_i \epsilon_j^k \quad (\text{B.1})$$

En introduisant un facteur de modulation S_j^m , la somme est alors calculée comme suit :

$$S_j^m = S_j^{m-1} \cdot \exp[-(t_j^m - t_j^{m-1})/\tau_p] + \epsilon_j^m \quad (\text{B.2})$$

Par conséquent la mise à jour du facteur de modulation à chaque nouveau PA en fonction du précédent évite de parcourir à chaque fois l'ensemble des anciens PAs. Il suffit alors de déterminer $S_j^m P(t_i - t_j^m) \epsilon_i$. Etant donné que $P(t) = A_p \exp(-t/\tau_p)$, nous retrouvons :

$$\begin{aligned}
& S_j^m P(t_i - t_j^m) \epsilon_i \\
&= [S_j^{m-1} \exp[-(t_j^m - t_j^{m-1})/\tau_p] + \epsilon_j^m] A_p \exp[-(t_i - t_j^m)/\tau_p] \epsilon_i \\
&= S_j^{m-1} P(t_i - t_j^{m-1}) \epsilon_i + P(t_i - t_j^m) \epsilon_i \epsilon_j^m \\
&= S_j^{m-2} P(t_i - t_j^{m-2}) \epsilon_i + P(t_i - t_j^{m-1}) \epsilon_i \epsilon_j^{m-1} + P(t_i - t_j^m) \epsilon_i \epsilon_j^m \\
&\dots \\
&= \sum_{k=0}^m P[t_i - t_j^k] \epsilon_i \epsilon_j^k
\end{aligned} \tag{B.3}$$

Nous procédons de la même manière avec la dépression, qui a lieu à chaque PA présynaptique au temps t_j , et qui est déterminée par l'ensemble des couples formés avec tous les PAs postsynaptiques aux temps t_i^k comme le présente l'équation B.4. Ainsi, nous utilisons pour effectuer le calcul un facteur de modulation S_i^m dont l'expression est fournie dans l'équation B.5.

$$\sum_{k=0}^m Q[t_j - t_i^k] \epsilon_i^k \epsilon_j \tag{B.4}$$

$$S_i^m = S_i^{m-1} \cdot \exp[-(t_i^m - t_i^{m-1})/\tau_q] + \epsilon_i^m \tag{B.5}$$

Ainsi, à chaque fois qu'un PA pré- ou postsynaptique est émis, l'historique des anciens PAs est pris en compte par la mise à jour du facteur de modulation correspondant. Cette méthode a l'avantage d'effectuer un minimum de calculs tout en considérant l'ensemble des PAs anciens.

Annexe C

Formats, programmes et scripts pour PAX et Gaillimh

Cette annexe présente dans un premier temps les formats utilisés pour stocker les données de configuration des systèmes et les données résultantes des simulations. Dans un deuxième temps, les différents programmes utilisés pour utiliser les systèmes sont décrits.

C.0.4 Formats des données

Dans les formats que nous allons décrire ci-dessous, nous utiliserons le caractère # pour indiquer qu'il s'agit d'un nombre décimal positif (sauf indication contraire) avec n'importe quel nombre de caractère.

Fichiers de configuration

Les formats de fichiers de configuration sont dans l'ensemble les mêmes pour les simulations avec PAX et Gaillimh. Tous les fichiers peuvent contenir des commentaires (qu'il faut faire débiter par le caractère '#') qui seront toujours ignorés.

- Fichiers avec l'extention ".it"

Ces fichiers contiennent les informations sur les neurones utilisés durant la simulation. Le format d'une ligne est :

```
[#] #, #, #;
```

Où le premier nombre entre crochets représente le numéro identifiant le neurone ([0-7] pour PAX et [0-12] pour Gaillimh), le second représente le type

du neurone (0 pour FS,1 pour RS), le troisième nombre représente le type de RS ([0-3] uniquement pour PAX sinon ignoré) et le dernier nombre représente la valeur numérique du courant de stimulation appliqué au neurone ([0-4095] uniquement pour PAX, sinon ignoré).

- Fichiers avec l'extention ".pds"

Ces fichiers contiennent les informations relatives aux connexions synaptiques. Le format est le suivant :

```
[p] #;
[m] #;

[# +/- #] #, #;
...
```

Les deux premières lignes ne concernent que PAX et restent présentes pour des raison de compatibilité avec Gaillimh. le nombre de la première ligne indique la précision du poids, la valeur optimale est 4. Le nombre de la seconde ligne indique le poids maximal ([0-255]).

Les lignes suivantes suivent toutes la structure de la troisième ligne où le premier nombre est le numéro identifiant du neurone présynaptique. Le signe désigne une synapse inhibitrice si "-" et une synapse excitatrice si "+"¹. Le deuxième nombre désigne le numéro du neurone postsynaptique. Le nombre suivant détermine le délai axonal (en *ms* et uniquement pour PAX, sinon ignoré doit être alors 0). Enfin le dernier nombre représente la valeur de poids initial de la synapse [0-255]. Ce nombre peut être négatif et dans ce cas signifie que la synapse n'est pas plastique (la synapse gardera la valeur absolue du poids tout au long de la simulation).

- Fichiers avec l'extention ".syn"

```
[p] #;

[#] #, #, #;
#, #, ..., #;

...
```

¹Notez que Gaillimh ignore le signe car il détermine en fonction du neurone présynaptique si la synapse est excitatrice ou inhibitrice.

Le nombre indiqué sur la première ligne indique le pas de temps utilisé (en μs). Les deux autres lignes définissent un pattern de bruit. La première des deux se lit ainsi : le premier nombre indique le neurone auquel on applique ce pattern de bruit. Le second nombre indique si le bruit est excitateur (1) ou inhibiteur (0). Le troisième nombre indique le poids de chaque stimulation de bruit (uniquement pour PAX sinon ignoré). Et le dernier nombre indique le nombre d'éléments (intervalle avant une stimulation) du pattern. La seconde ligne indique les intervalles entre deux stimulations de bruit. Il y a autant de nombres qu'indiqué dans le dernier paramètre de la ligne précédente.

- Fichiers avec l'extension ".prj" pour PAX

Un fichier ".prj" détermine la configuration de la simulation. Le format est le suivant :

```
[nbn]#;
[pds] "CHEMIN/VERS/NOM_DU_FICHER.pds";
[it] "CHEMIN/VERS/NOM_DU_FICHER.it";
[syn] "CHEMIN/VERS/NOM_DU_FICHER.syn";
```

Le nombre à indiquer sur la première ligne indique le nombre de neurones utilisés pour la simulation. les trois lignes indiquent respectivement les noms de fichier de configuration ".it", ".pds" et ".syn" pour la simulation.

- Fichiers avec l'extension ".prj" pour Gaillimh

Un fichier ".prj" détermine la configuration de la simulation. Le format est le suivant :

```
CHEMIN/VERS/NOM_DU_FICHER.it
CHEMIN/VERS/NOM_DU_FICHER.pds
CHEMIN/VERS/NOM_DU_FICHER.syn
```

les trois lignes indiquent respectivement les noms de fichier de configuration ".it", ".pds" et ".syn" pour la simulation. Contrairement au fichier ".prj" pour PAX, le nombre de neurones est déterminé par les neurones listés dans le fichier ".it".

- Fichiers avec l'extension ".ini" pour Gaillimh

Ce fichier contient les valeurs des paramètres analogiques pour configurer les circuits Galway. Ces fichiers ont été obtenus suite au réglage des neurones décrit dans le chapitre 3. L'optimisation des paramètres analogiques devraient fournir des fichiers de ce type. Le fichier est constitué de 6 colonnes

sur 48 lignes de nombre réels. Chaque nombre a une précision de 3 chiffres après la virgule. Certains nombres sont à zéro car inutilisés. Le nombre total de paramètres utiles est de 205. Le tableau C.1 présente la signification de chacun des 205 paramètres analogiques du circuit. Ils sont numérotés de 0 à 204 sous forme hexadécimale, ce qui permet d'effectuer directement le lien entre le paramètre et une adresse mémoire qui lui correspond. Ce tableau est extrait de l'annexe E de la thèse de Yannick Bornat [79].

Signification du paramètre	Neurone					Cond. libre	
	0	1	2	3	4		
Na^+ -act- V_{pente}	00	1E	45	6C	93		
Na^+ -act- V_{offset}	01	1F	46	6D	94		
Na^+ -act- τ	02	20	47	6E	95		
Na^+ -inact- V_{pente}	03	21	48	6F	96		
Na^+ -inact- V_{offset}	04	22	49	70	97		
Na^+ -inact- τ	05	23	4A	71	98		
Na^+ - g_{max}	06	24	4B	72	99		
Na^+ - V_{equi}	07	25	4C	73	9A		
K^+ - V_{pente}	08	26	4D	74	9B		
K^+ - V_{offset}	09	27	4E	75	9C		
K^+ - τ	0A	28	4F	76	9D		
K^+ - g_{max}	0B	29	50	77	9E		
K^+ - V_{equi}	0C	2A	51	78	9F		
Fuite - g_{max} (1/2)	0D	2B	52	79	A0		
Fuite - g_{max} (2/2)	0E	2C	53	7A	A1		
Fuite - V_{equi}	0F	2D	54	7B	A2		
Ca^{2+} -act- V_{pente}		2E	55	7C	A3		
Ca^{2+} -act- V_{offset}		2F	56	7D	A4		
Ca^{2+} -act- τ		30	57	7E	A5		
Ca^{2+} -inact- V_{pente}		31	58	7F	A6		
Ca^{2+} -inact- V_{offset}		32	59	80	A7		
Ca^{2+} -inact- τ		33	5A	81	A8		
Ca^{2+} - g_{max} (1/2)		34	5B	82	A9		
Ca^{2+} - g_{max} (2/2)		35	5C	83	AA		

TAB. C.1: Correspondance entre les paramètres et leur numéro d'ordre (continué page suivante)

Signification du paramètre	Neurone					Cond. libre	
	0	1	2	3	4		
$Ca^{2+} - V_{equi}$		36	5D	84	AB		
mod. -act- V_{pente}					AC	C3	
mod. -act- V_{offset}					AD	C4	
mod. -act- τ					AE	C5	
mod. -inact- V_{pente}					AF	C6	
mod. -inact- V_{offset}					B0	C7	
mod. -inact- τ					B1	C8	
mod. - g_{max} (1/2)					B2	C9	
mod. - g_{max} (2/2)					B3	CA	
mod. - V_{equi}					B4	CB	
Syn. A - τ	10	37	5E	85	B5		
Syn. A - g_{max} (1/2)	11	38	5F	86	B6		
Syn. A - g_{max} (2/2)	12	39	60	87	B7		
Syn. A - V_{equi}	13	3A	61	88	B8		
Syn. B - τ	14	3B	62	89	B9		
Syn. B - g_{max} (1/2)	15	3C	63	8A	BA		
Syn. B - g_{max} (2/2)	16	3D	64	8B	BB		
Syn. B - V_{equi}	17	3E	65	8C	BC		
Syn. C - τ	18	3F	66	8D	BD		
Syn. C - g_{max} (1/2)	19	40	67	8E	BE		
Syn. C - g_{max} (2/2)	1A	41	68	8F	BF		
Syn. C - V_{equi}	1B	42	69	90	C0		
Stimulation (1/2)	1C	43	6A	91	C1		
Stimulation (2/2)	1D	44	6B	92	C2		
Caractérisation							CC

TAB. C.1: Correspondance entre les paramètres et leur numéro d'ordre

Fichiers de résultats bruts

Les fichiers générés par les programmes de simulation de PAX et Gaillimh sont présentés ici.

- Fichier de PAs (extension ".spk")

, #;

Chaque ligne de ce fichier possède deux informations. Le premier nombre est le numéro de neurone ayant émis un PA. Le second est un réel indiquant le temps d'émission en secondes. En général la première ligne du fichier est un commentaire indiquant la nature des deux informations ("*#neur t(s);*")

- Fichier de poids (extension ".w")

Ce fichier stocke les informations concernant l'évolution des poids synaptiques soumis à la plasticité.

#, #, #, #, #;

Chaque ligne de ce fichier contient 5 informations. Les deux premiers nombres représentent respectivement les numéros du neurone présynaptique et du neurone postsynaptique. Le troisième nombre représente le type de la synapse (1 pour excitatrice, 0 pour inhibitrice), le nombre suivant indique le temps auquel le calcul de plasticité a eu lieu. Enfin le dernier nombre indique la nouvelle valeur du poids calculée. En général la première ligne du fichier est un commentaire indiquant la nature des informations ("*#pre, post, type, t, w;*")

- Fichier récapitulatif (extension ".res")

Ce fichier est généré uniquement avec PAX. Il reprend le contenu des fichiers de configuration (fichiers avec les extensions ".it", ".pds", ".syn", et ".prj").

- Fichier du bruit (extension ".snoi")

Ce fichier présente différemment les patterns de bruit. Il les présente sous une forme inspirée du fichier avec l'extension ".spk". Chaque ligne est constituée comme suit :

#, #, #;

Le premier nombre représente le numéro du neurone ayant été stimulé. Le second nombre indique le type de la synapse stimulée (0 pour inhibitrice, 1 pour excitatrice). Enfin le troisième nombre indique le temps en secondes où la synapse a été stimulée.

Fichiers de résultats traités

Les données brutes sus-citées, sont ensuite traitées par les scripts décrits dans le tableau C.5. En général les fichiers traités ont pour nom un préfixe "TRI" et un suffixe dépendant du traitement. Dans la suite nous utiliserons les noms génériques "fic.spk" et "fic.w" comme noms des fichiers de données brutes. Les données traitées sont toujours écrites de façon simple sans les caractères ajoutés aux données brutes (tels ",", ";" et "[]"). Les seuls séparateurs utilisés sont la tabulation et le retour à la ligne. Deux retours à la ligne consécutifs sont utilisés pour séparer des séries de données. Cette organisation des données facilite le traitement par Gnuplot. Le tableau C.2 fournit les différents noms des fichiers et un petit descriptif.

Fonction	Description
1. TRI <i>fic.spk</i> .spk.plot	Données de PAs triés par neurone et ordre croissant temporel (neurone, temps)
2. TRI <i>fic.spk</i> .freq.plot	Fréquences instantanées (temps, fréq)
3. TRI <i>fic.spk</i> .fm.plot	Fréquence moyenne générale
4. TRI <i>fic.spk</i> .data.plot	Fréq. moyenne par neurone, fréq. moy. instantanée par neurone et écart-type (neurone, fm, fmi, ET)
5. TRI <i>fic.w</i> .plot	Poids triés par synapse (temps, poids)
6. TRI <i>fic-dt#-nbt#</i> .ppspk.plot	Histogramme de corrélation des PAs (x,y)
7. TRI <i>fic-dw#</i> .pw.plot	Histogramme de répartition de poids (x,sum,y)
8. TRI <i>fic.spk-dt#</i> .matco.plot	Histog. de corrélation à différents moments de la simulation
9. TRI <i>fic.w-dw#</i> .paramW.plot	Catégorisation des poids avec PearsonVII (pf,pF)
10. TRI <i>fic.w</i> .mw.plot	Fichier de matrices de valeurs de poids d'une simulation à différents moments (pre VS post)
11. nom contenant <i>corri-coor</i>	Corrélations des PAs en fonction de la corrélation du bruit
12. nom contenant <i>coall</i>	Type de répartition des poids en fonction de la corrélation du bruit

TAB. C.2 – Liste des résultats traités générés à partir des données brutes "fic.spk" et "fic.w".

C.0.5 Programmes et scripts

Les programmes et scripts sont accompagnés de la GNU General Public Licence v3. Le code est commenté en fonction des besoins. La plupart des commentaires sont en anglais afin de faciliter la réutilisation des programme par des non-francophones. Veuillez vous référer au code source pour avoir des informations plus précises sur la fonction des programmes et scripts. La plupart d'entre eux requièrent des arguments, et le fait de les exécuter sans arguments ou si le nombre d'arguments est insuffisant, affichera une aide précisant la syntaxe d'utilisation. Pour les programmes en C, il sera plus simple de se référer aux fichiers d'entêtes ".h" dans le dossier *include*. Ils contiennent en général la liste des fonctions du programme C qui correspond et un bref descriptif.

Programmes de simulation et de configuration

Les simulateurs PAX et Gaillimh ont des programmes les exploitant bien distincts. Les programmes pour Gaillimh devraient pouvoir être réutilisés avec peu de modifications pour exploiter le système en préparation utilisant les puces Galway. Nous ne présentons ici que les programmes nécessaires pour effectuer des simulations automatisées. D'autres programmes ont été développés afin de réaliser des tests, de réaliser des courbes f(I) et ISI, des configurations en pas par pas, et d'autres opérations.

- Programme de simulation pour PAX

L'exécutable *pax_sim* est compilé à partir de plusieurs fichiers au format C. Le tableau C.3 fournit un descriptifs de ces fichiers. Un fichier *Makefile* permet d'effectuer la compilation. Pour lancer des simulations, il faut utiliser le script bash *simul* qui lui-même appelle le programme *pax_sim*.

- Programme de simulation pour Gaillimh

L'exécutable *gail_sim* est compilé à partir de plusieurs fichiers au format C. Le tableau C.4 fournit un descriptifs de ces fichiers. Un fichier *Makefile* permet de le générer.

Scripts de traitement des données

Les résultats des simulations sont dans un premier temps traités avant de pouvoir générer des graphiques. Nous présentons ici les principaux scripts permettant de traiter le données et de les préparer à être transformés en

Fonction	Description
stdprep.c	Programme principal exécutant la simulation
stdp.c	Fonctions initialisant et calculant la STDP
pax2.c	Fonctions gérant le protocole de configuration et de communication avec le FPGA via le PCI
nvsauvegarde.c	Fonctions permettant de lire les fichiers de configuration
lectconf.c	Fonctions permettant de décomposer le format des fichiers de configuration
manipoids.c	Fonctions gérant la liste chaînée contenant les informations relatives aux synapses
manisyn.c	Fonctions gérant la liste chaînée contenant les informations relatives aux patterns de bruit
tire_poisson.c	Génère les patterns de bruit corrélés (lancer sans arguments pour l'aide)
tire_poisson_sg.c	Génère les patterns de bruit corrélés tenant compte du groupe de 2 groupes de neurones

TAB. C.3 – Liste des programmes C composant l'exécutable *pax_sim*, et programmes de génération des fichiers de patterns de bruit synaptiques d'extension ".syn"

Fonction	Description
main_sim.c	Programme principal exécutant la simulation
general.c	
fct_bas-niveau.c	Fonctions gérant la communication avec le FPGA via ports série
fct_haut-niveau.c	Fonctions gérant le protocole de configuration du FPGA
fromfile.c	Fonctions permettant de lire les fichiers de configuration
lectconf.c	Fonctions permettant de décomposer le format des fichiers de configuration
manipoids.c	Fonctions gérant la liste chaînée contenant les informations relatives aux synapses
nz_sim.c	Programme principal remplaçant main_sim.c et gérant le bruit
gere_bruit.c	Fonctions permettant la gestion du bruit
manisyn.c	Fonctions gérant la liste chaînée contenant les informations relatives au bruit

TAB. C.4 – Liste des programmes C composant l'exécutable *gail_sim*.

graphiques par Gnuplot. Le tableau C.5 fournit la liste de ces scripts et un petit descriptif pour chacun. Veuillez vous référer aux scripts pour avoir des détails notamment concernant les arguments que nécessitent les différents scripts.

Fonction	Description
freqmoyet.pl	Traite les données brutes des PAs et génère les fichiers 1 à 4 tableau C.2
triwplotp.pl	Traite les données brutes d'évolution des poids (fichier 5)
tricorrel.pl	Génère les histogrammes de corrélation (fichier 6)
tridistriw.py	Génère les histogrammes de poids (fichier 7)
matr_corr.pl	Génère l'évolution de la corrélation des PAs dans une simulation (fichier 8)
calc_paramW.pl	Calcul le type de répartition avec les équations de Pearson VII (fichier 9)
genere_pw.pl	Applique les 4 derniers scripts au dossier courant
dernpds.pl	Génère des matrices de poids d'une simu. (fichier 10)
trace-mw.pl	Affiche avec SDL les matrices générées par dernpds.pl ESPACE affiche la matrice suivante, ECHAP termine le programme
rassemb-corr.pl	Regroupe les données de plusieurs fichiers 6 du tableau C.2 (fichier 11)
rassemb-pW.pl	Regroupe les données de plusieurs fichiers 9 (fichier 12)

TAB. C.5 – Liste des scripts permettant d'effectuer les différents traitements sur les données de simulation.

Scripts de génération de graphiques

Les scripts permettant de générer les graphiques, sont des scripts Gnuplot incorporés dans des scripts bash. Le tableau C.6 fournit le nom de ces scripts et un petit descriptif. La plupart du temps les scripts génèrent des fichiers au format PNG ou EPS.

Fonction	Description
genplot_swf_onp	Génère un ensemble de 4 graphiques comportant l'évolution de la fréquence instantanée des neurones, l'évolution des poids, 2 diagrammes de PAs utilise les fichiers 1, 2 et 3 du tableau C.2
genplot_moyet	Génère un graphique de fréq. moyenne, fréq. moy. instantanée et ecart-types (fichier 4)
genplot_pwsbin	Génère les 2 graphiques d'histogrammes de poids et de corrélation (fichiers 6 et 7)
genplot_matrcorr	Génère un graphique montrant l'évolution de la corrélation des PAs des neurones durant la simulation (fichier 8)
genplot_paramW	Génère un graphique avec un point représentant le type de répartition des poids (fichier 9)
genplot_corr	Génère un graphique présentant la corrélation des PAs en fonction de la corrélation du bruit (fichier 11)
genplot_paramW_multi	Idem que genplot_paramW mais pour plusieurs corrélations du bruit (fichier 12)

TAB. C.6 – Liste des scripts qui génèrent les graphiques.

Annexe D

Protocole de communication pour Gaillimh

Cette section présente le protocole de communication, entre le PC et le FPGA, implanté sur la carte Gaillimh.

D.0.6 Attribution des ports série

- Transmission PC -> FPGA :

- >Port STD : envoi des paramètres analogiques/numériques
- >Port 0 : envoi des paramètres du réseau (connectivité)
- >Port 1 : envoi des informations concernant le bruit synaptique

- Transmission FPGA -> PC :

- >Port STD : informations concernant le bruit synaptique (pour sa mise à jour)
- >Port 0 : envoi informations PAs/poids
- >Port 1 : envoi informations PAs/poids

D.0.7 Paramètres analogiques et numériques

La communication avec le FPGA s'effectue par l'échange de chaînes de caractères. Nous allons nous placer du point de vue de l'ordinateur, ainsi l'envoi d'une chaîne concernera la transmission de la chaîne du programme informatique au FPGA, et la réception désignera la transmission inverse.

L'envoi d'instructions et de données se fait par des caractères ASCII ayant la forme générique suivante $X##$. Le X est une lettre de l'alphabet désignant le type d'opération et le $\#$ est un caractère hexadécimal représentant une valeur à transmettre¹. En fonction de l'instruction le nombre de caractères hexadécimaux peut varier et même être nul.

Initialisation

L'initialisation du système se fait en envoyant deux mots d'instruction à travers le port 0 :

'R' pour le Reset

'S' pour le début de simulation

Envoi des paramètres analogiques

Les paramètres analogiques correspondent aux valeurs des paramètres de configuration des neurones selon le modèle de Hodgkin et Huxley adaptés aux ASICs Galway.

Trame d'envoi² :

$A\# \#\# \#\#\#\#$

Les sous-composants de cette chaîne sont :

- 'A' indique que l'opération concerne les paramètres analogiques.

- Le chiffre hexadécimal qui le suit désigne le numéro de l'ASIC concerné (de 1 à 5).

- Les 2 chiffres suivants indiquent l'adresse du paramètre dans l'ASIC. La valeur de cette adresse est comprise entre 0x00 et 0xCC. Le tableau C.1 de l'annexe C fournit la correspondance. Par exemple l'adresse 0x9E concerne la conductance potassique \bar{g}_K du neurone numéro 4 de l'ASIC.

- Les 4 derniers chiffres représentent la valeur du paramètre sous forme d'un nombre entier sur 2 octets. Le FPGA code la valeur du paramètre sur 14 bits. Par conséquent la valeur maximale est 0x3FFF. Pour le système les valeurs des paramètres du modèle HH sont des valeurs de tension comprises entre 0 et 5 V représentées par un nombre réel. Or la chaîne envoyée transmet un nombre entier. L'équation D.1 fournit la conversion de la tension

¹Précisons que deux '#' fournissent les 4 bits de poids fort et 4 bits de poids faible d'un octet.

²En format pour la fonction *printf()* en langage C cela donne "A%d%2.2X%4.4X".

électrique u en entier val traité par le FPGA. L'arrondi se fait à l'entier le plus petit supérieur ou égal (fonction $ceil()$ en langage C).

$$val = arrondi(u * 0x3FFF/5, 0) \quad (D.1)$$

Les paramètres pourront être envoyés un par un ou en série, sans importance quant à l'ordre. Pour valider l'envoi du paramètre ou de la série, il est nécessaire de terminer par l'envoi de la chaîne de caractères suivante : **A0##0000**.

où ## désigne le numéro de l'ASIC.

Envoi des paramètres numériques

Les paramètres numériques concernent l'organisation à l'intérieur d'un ASIC. Il permettent d'indiquer quelles conductances sont connectées à la membrane des neurones ou à la résistance de visualisation, le mode de fonctionnement (*voltage-clamp* ou *simulation*) et le numéro du neurone concerné.

Trame d'envoi³ :

D# ## ##

Les sous-composants sont :

- 'D' indique que l'opération concerne les paramètres numériques.

- Le chiffre suivant est le numéro de l'ASIC concerné.

- Les 4 chiffres hexadécimaux suivants indiquent la valeur du paramètre numérique codé sur les 14 bits de poids faible des 2 octets. La signification des différents bits est fournie par les tableaux E.1, E.2, E.3, E.4 et E.5 de l'annexe E de la thèse de Y. Bornat⁴

D.0.8 Gestion du bruit synaptique

Généralités

La gestion du bruit synaptique se fait de façon particulière. De plus elle ne peut fonctionner que pour un maximum de 16 neurones. Nous avons défini cette limite pour des raisons techniques et parce que nous disposons de moins de 16 neurones fonctionnels. Pour chaque type de synapse de chaque neurone,

³en langage C : " $D\%d\%4.4X$ ".

⁴Attention, les bits sont numérotés de 0 à 13 en commençant par le bit de poids fort (donc numéro 0 pour le bit d'indice 13 du mot de 2 octets et 13 pour le bit de poids faible).

est attribué un emplacement mémoire permettant de stocker 256 entiers de 16 bits chacun. Chaque valeur qui y sera stockée, représente l'intervalle de temps (en millisecondes), qu'on appelle aussi ISI⁵, avant la prochaine stimulation de bruit synaptique. Le FPGA, décrémente pour chaque neurone l'ISI en cours de lecture à chaque pas de temps, et quand le ISI devient nul, le FPGA stimule l'entrée synaptique du neurone, et passe à l'ISI à traiter suivant pour ce neurone. Quand pour un neurone donné le FPGA atteint le dernier ISI de la mémoire, il recommence à la première case. Il est donc nécessaire, pour utiliser plus de 256 ISIs, de fournir en cours de traitement les nouveaux ISIs stockés sur l'ordinateur. Par conséquent, le programme informatique doit envoyer en permanence au FPGA des données de mise-à-jour des patterns de bruit. Le traitement global est comme suit. Le programme envoie pour tous les compartiments les 256 valeurs de ISI, puis un message de fin de transmission et attend une réponse du FPGA qui indique l'adresse d'ISI en cours de traitement d'un compartiment dont des données ont été traitées. A ce moment-là, le programme doit envoyer les nouveaux ISIs qui remplaceront ceux déjà lus et se rappellera de l'adresse de traitement actuel afin d'envoyer la prochaine fois les données à partir de cette adresse et jusqu'à la nouvelle adresse qui sera alors lue. La valeur maximale d'un ISI (codé sur 16 bits) est 65535 ms. Notez qu'il n'est pas possible de désactiver le traitement des patterns de bruit. Ceci implique qu'il faut initialiser tous les ISIs à la valeur maximale pour s'affranchir du bruit dont cependant une stimulation toutes les 65,5 secondes aura lieu.

Pour la gestion du bruit, le FPGA utilise 8 mémoires (RAM), numérotés de 0 à 7, qu'il répartit par couple de neurones. Ainsi une RAM est allouée pour 2 neurones et est décomposée en 4 compartiments (numérotés de 0 à 3), comme le montre la figure D.1. Les compartiments 0 et 1 concernent le bruit synaptique respectivement excitateur et inhibiteur du neurone 0, et les compartiments 2 et 3 du neurone 1.

Protocole de gestion du bruit

Nous rappelons que pour le bruit l'envoi des paramètres du PC vers le FPGA se fait par le port P1 et la réception par le port STD.

Transmission PC->FPGA

⁵Inter-Spike Interval

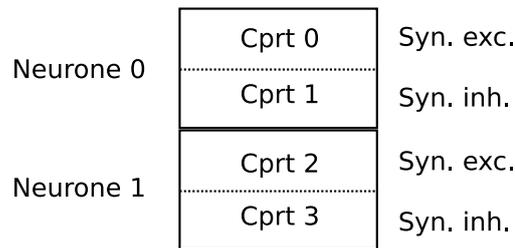


FIG. D.1 – Organisation d'une RAM pour la gestion du bruit synaptique.

Trame d'envoi :

N## N## N## N##

Sous-composition :

- 'N' signifie qu'il s'agit d'opérations en lien avec la gestion du bruit
- Le chiffre suivant indique le numéro de la RAM [0-7].
- Le chiffre suivant indique le numéro de compartiment [0-3].
- Les 2 chiffres suivants indiquent l'adresse de l'ISI [0-255].
- Enfin les 4 derniers chiffres indiquent la valeur de l'ISI [0-65535].

Enfin, pour indiquer la fin de la transmission, il faut envoyer le caractère 'Z'. Cette instruction va entraîner une réponse du FPGA, respectant la trame ci-dessous, correspondant à l'adresse à laquelle envoyer de nouvelles données.

Transmission FPGA->PC

Trame de réception :

##

Sous-composition :

- Le premier chiffre indique le numéro de la RAM [0-7].
- Le second chiffre indique le numéro de compartiment.
- Les 2 derniers chiffres indiquent l'adresse de l'ISI.

D.0.9 Réception des données

La réception des données de simulation se fait alternativement sur le port P1 puis P0. Les données reçues peuvent concerner l'horloge, un événement PA ou un événement changement de poids. La taille de la trame est de 10 caractères hexadécimaux.

Les informations concernant l'horloge, dont le pas est la milliseconde, a le même format que pour un neurone. Etant donné que seuls 13 neurones fonctionnent sur Gaillimh, nous avons associé le numéro de neurone 15 à l'horloge. Le temps est cependant codé sur 23 bits pour les trames de réception des informations concernant les poids, et sur 31 bits concernant les PAs⁶

Gestion des PAs

Trame de réception :

#####

Sous-composition :

-Les 2 premiers chiffres représentent le numéro de neurone [0-255] (seuls les neurones [0-12] fonctionnent). Si le numéro de neurone est 15 (0x0F) cela désigne l'évènement horloge.

-Les 8 chiffres suivants fournissent une valeur sur 32 bits dont les 31 bits de poids fort correspondent au temps d'émission du PA.

-Le bit de poids faible est à '0' pour indiquer que les informations reçues concernent les PAs.

Gestion des poids

Trame de réception :

#####

Sous-composition :

-Le premier chiffre indique le numéro de neurone pré-synaptique [0-16]

-Le second chiffre indique le numéro de neurone post-synaptique [0-16]

-Les 2 chiffres suivants représente la valeur du poids calculée par STDP [0-255]

-Les 6 derniers chiffres fournissent une valeur sur 24 bits dont les 23 bits de poids fort donnent le temps du calcul du poids.

⁶Il faut veiller à ce que le programme calcule le temps réel depuis le début de la simulation à partir de compteurs distincts pour les poids et pour les PAs (des neurones et de l'horloge).

-Le bit de poids faible est à '1' pour indiquer que les informations reçues concernent les poids synaptiques.