

Thèse de Doctorat

de l'Université de Bordeaux

École doctorale des Sciences Physiques et de l'Ingénieur

Spécialité Electronique

Préparée au Laboratoire IMS

Par Matthieu Ambroise

Hybridation des réseaux de neurones : De la conception du réseau à l'interopérabilité des systèmes neuromorphiques

Soutenue le 7 Décembre 2015

Après avis de :

M. Jean Marie BILBAULT
M. Benoit MIRAMOND

Professeur des Universités - Université de Bourgogne
Professeur des Universités - Université de Nice Sophia Antipolis

Devant la commission d'examen formée de:

M. Jean Marie BILBAULT
M. Sio-Hoi IENG
M. Timothée LEVI
M. Benoit MIRAMOND
M. Claude PELLET
M. Sylvain SAÏGHI

Professeur des Universités -Université de Bourgogne
Maître de Conférences - Université Pierre et Marie Curie
Maître de Conférences - Université de Bordeaux
Professeur des Universités -Université de Nice Sophia Antipolis
Professeur des Universités - Université de Bordeaux
Maître de Conférences HDR - Université de Bordeaux

Rapporteur
Examinateur
Co-directeur de thèse
Rapporteur
Président du jury
Directeur de thèse



*À mes parents,
À mon frère,
À ma soeur.*

Thèse réalisée
dans le laboratoire **IMS**
au sein de l'équipe **AS2N**.

Université de Bordeaux
Laboratoire IMS
CNRS UMR-5218
351, cours de la Libération
33405 TALENCE Cedex

REMERCIEMENTS

JE tiens en tout premier lieu à remercier M. Claude Pellet pour m'avoir accueilli dans le laboratoire IMS pendant la réalisation de ces travaux de thèse.

Mes sincères remerciements vont à M. Sylvain Saïghi et M. Timothée Lévi qui ont dirigé cette thèse et m'ont encadré durant trois années. Merci pour votre soutien, l'aide précieuse et les conseils prodigues. Il est agréable de travailler à vos côtés. Vous m'avez permis de découvrir un monde scientifique d'une beauté insoupçonnée et par-dessus tout vous m'avez fait "grandir". Je suis heureux d'avoir pu aider l'équipe à travailler dans une nouvelle direction.

Je tiens à remercier chaleureusement M. Jean-Marie Bilbault et M. Benoit Miramond, rapporteurs de cette thèse, pour leurs lectures, leurs commentaires sur ce manuscrit et également pour le temps qu'ils m'ont accordé.

Je remercie également M. Sio-Hoi IENG d'avoir consacré du temps à examiner mes travaux de thèse.

Je remercie encore une fois les membres du jury d'avoir accepté d'être dans ce jury.

Mes remerciements vont également à M. Arnaud Languet et à M. Aswhin Mangalore, collègues ingénieurs qui ont brillé tant par leur savoir-faire que leur bonne humeur. Je remercie également mon stagiaire M. Charles Bailly qui est venu travailler à mes côtés. Je remercie également les équipes biologistes avec lesquelles j'ai pu collaborer et plus particulièrement Mme Michela Chiappalone et son équipe, M. Blaise Yvert, M. Sébastien Joucla.

Je remercie l'ensemble des personnes de l'IMS, le personnel de soutien à la recherche indispensable au bon fonctionnement du laboratoire. Parmi le personnel, je remercie Valérie et Simone.

Je voudrais également remercier l'ensemble des collègues du groupe bioélectronique (les équipes AS2N, BioEM et Elibio) : Jean, Sylvie, Noëlle, Yannick, Gilles, François, Gwendal, Florian, Fillipo, Jean-Baptiste, Adam, Antoine, Jonathan, Hermanus, Arnaud, Adeline. Merci aussi à mes amis proches et aux jeunes du GDJ.

Je remercie également l'ensemble des membres de ma famille, mon père, ma mère, Marie-Céline et sa famille, Christian et sa famille ! Vous m'avez toujours soutenu, je voulais vous dédier ce travail.

RÉSUMÉ

L'hybridation est une technique qui consiste à interconnecter un réseau de neurones biologique et un réseau de neurones artificiel, utilisée dans la recherche en neuroscience et à des fins thérapeutiques. Durant ces trois années de doctorat, ce travail de thèse s'est focalisé sur l'hybridation dans un plan rapproché (communication directe bi-directionnelle entre l'artificiel et le vivant) et dans un plan plus élargies (interopérabilité des systèmes neuromorphiques).

Au début des années 2000, cette technique a permis de connecter un système neuromorphe analogique avec le vivant. Ce travail est dans un premier temps, centré autour de la conception d'un réseau de neurones numérique, en vue d'hybridation, dans deux projets multi-disciplinaires en cours dans l'équipe AS2N de l'IMS, présentés dans ce document :

- *HYRENE* (ANR 2010-Blan-031601), ayant pour but le développement d'un système hybride de restauration de l'activité motrice dans le cas d'une lésion de la moelle épinière,
- *BRAINBOW* (European project FP7-ICT-2011-C), ayant pour objectif l'élaboration de neuro-prothèses innovantes capables de restaurer la communication autour de lésions cérébrales.

Possédant une architecture configurable, un réseau de neurones numérique a été réalisé pour ces deux projets. Pour le premier projet, le réseau de neurones artificiel permet d'émuler l'activité de CPGs (Central Pattern Generator), à l'origine de la locomotion dans le règne animal. Cette activité permet de déclencher une série de stimulations dans la moelle épinière lésée *in vitro* et de recréer ainsi la locomotion précédemment perdue. Dans le second projet, la topologie du réseau de neurones sera issue de l'analyse et le décryptage des signaux biologiques issus de groupes de neurones cultivés sur des électrodes, ainsi que de modélisations et simulations réalisées par nos partenaires. Le réseau de neurones sera alors capable de réparer le réseau de neurones lésé.

Ces travaux de thèse présentent la démarche de conception des deux différents réseaux et des résultats préliminaires obtenus au sein des deux projets.

Dans un second temps, ces travaux élargissent l'hybridation à l'interopérabilité des systèmes neuromorphiques. Au travers d'un protocole de communication utilisant Ethernet, il est possible d'interconnecter des réseaux de neurones électroniques, informatiques et biologiques. Dans un futur proche, il permettra d'augmenter la complexité et la taille des réseaux.

Mots Clés :

FPGA, Réseau de neurones impulsionnels, CPG, Hybridation, AER, Protocole de communication.

ABSTRACT

HYBRID experiments allow to connect a biological neural network with an artificial one, used in neuroscience research and therapeutic purposes. During these three years of PhD, this thesis focused on hybridization in a close-up view (bi-directionnal direct communication between the artificial and the living) and in a broader view (interoperability of neuromorphic systems).

In the early 2000s, an analog neuromorphic system has been connected to a biological neural network. This work is firstly, about the design of a digital neural network, for hybrid experiments in two multi-disciplinary projects underway in AS2N team of IMS presented in this document :

- *HYRENE* (ANR 2010-Blan-031601), aiming the development of a hybrid system for the restoration of motor activity in the case of a spinal cord lesion,
- *BRAINBOW* (European project FP7-ICT-2011-C), aiming the development of innovative neuro-prostheses that can restore communication around cortical lesions.

Having a configurable architecture, a digital neural network was designed for these two projects. For the first project, the artificial neural network emulates the activity of CPGs (Central Pattern Generator), causing the locomotion in the animal kingdom. This activity will trigger a series of stimuli in the injured spinal cord *textit{in vitro}* and recreating locomotion previously lost. In the second project, the neural network topology will be determined by the analysis and decryption of biological signals from groups of neurons grown on electrodes, as well as modeling and simulations performed by our partners. The neural network will be able to repair the injured neural network.

This work show the two different networks design approach and preliminary results obtained in the two projects.

Secondly, this work hybridization to extend the interoperability of neuromorphic systems. Through a communication protocol using Ethernet, it is possible to interconnect electronic neural networks, computer and biological. In the near future, it will increase the complexity and size of networks.

Keywords:

FPGA, Spiking Neural Network, CPG, Hybrid experiments, AER, Protocole de communication.

Abréviations

AdEx *Adaptive Exponential Integrate & Fire*

AS2N *Architecture of Silicon Neural Networks*

BIOEM *BIOElectroMagnétisme*

CPG *Central Pattern Generator*

Elibio *Systèmes ELectroniques en Interaction avec la BIOlogie*

ESIEE *Ecole Supérieure d'Ingénieurs en Electronique*

FPGA *Field Programmable Gate Array*

GIHON *Generic Interconnection between Heterogeneous spiking neurONs*

HH *Hodgin-Huxley*

HYRENE *HYbridation de REseaux de NEurones pour l'exploration de fonctions de réhabilitation*

IF *Integrate & Fire*

IIT *Institut Italien de Technologie*

IMS *Intégration du Matériaux au Système*

INCIA *Institut de Neurosciences Cognitives et Intégratives*

IZH *Izhikevich*

LIF *Leaky Integrate & Fire*

MCS *MultiChannel Systems*

MEA *Multi ou Micro-Electrode Array*

PA potentiel d'action

TAU *Université de Tel-Aviv*

UNIGE *UNIversité de GÊne*

VHDL *VHSIC (Very High Speed Integrated Circuit) Hardware Description Language*

TABLE DES MATIÈRES

Remerciements	vii
Résumé	xi
Abstract	xv
Abréviations	xvii
Table des matières	xxi
1 De l'anatomie du neurone aux systèmes neuromorphiques	3
1.1 Morphologie et physiologie d'un neurone biologique	5
1.2 Physiologie succincte du neurone	7
1.3 Principes de modélisation neuromorphique	10
1.3.1 Modèle Hodgkin-Huxley (HH)	11
1.3.2 Modèle "Integrate and fire" (IF) et modèle "Leaky Integrate and Fire" (LIF)	13
1.3.3 Modèle Izhikevich (IZH)	14
1.3.4 Modèle "Adaptive Exponential integrate and fire" (Adex)	15
1.3.5 Modèle quartique (Tou)	16
1.3.6 Le meilleur modèle de neurone pour nos applications	16
1.4 Brève présentation de modèles de synapses	17
1.5 De l'implémentation de réseaux vers les réseaux de plateformes	18
1.5.1 Qu'est ce qu'un réseau ?	18
1.5.2 Etat de l'art des implémentations neuromorphiques	18
1.5.3 Les réseaux dans l'équipe AS2N (Bordeaux)	22
1.6 Conclusion	23
2 Implémentation de réseau de neurones sur FPGA	27
2.1 Principes de fonctionnement d'un FPGA	29
2.1.1 Le FPGA	29
2.1.2 Le VHDL : VHSIC Hardware Description Language	30
2.2 Cahiers des charges hérités des projets BRAINBOW et HYRENE	30
2.2.1 Cahier des charges BRAINBOW	31
2.2.2 Cahier des charges HYRENE	32
2.3 Représentation des données	32
2.3.1 Principe de la virgule flottante	32
2.3.2 Principe de la virgule fixe	33
2.3.3 Quantification de la correspondance entre la représentation en virgule fixe et la représentation en virgule flottante	33
2.4 Architecture de notre implémentation	35
2.4.1 Coeur de calcul de neurones	35
2.4.2 Coeur de calcul de synapses	40
2.4.3 Rôle de la RAM	42
2.4.4 Gestion des calculs au sein de notre système	45
2.4.5 Reconfigurabilité du réseau via une liaison série	45
2.4.6 Performances du réseau	51
2.5 Conclusion	53

3 Mise en œuvre de notre réseau et Expériences hybrides	55
3.1 Utilisation du réseau de neurones	58
3.1.1 Réseau "Chenillard"	58
3.1.2 Réseau "Chenillard" plastique	59
3.1.3 Réseau "Reconnaissance de motifs temporels"	59
3.2 Théorie des Central Pattern Generator	61
3.2.1 Approche connexionniste	61
3.2.2 Principe de l'implémentation matérielle	62
3.2.3 Passage du CPG Hodgkin-Huxley au CPG Izhikevich	62
3.2.4 Topologie des CPGs	64
3.2.5 Réglage de la période des bouffées de potentiels d'actions	67
3.3 Résultats expérimentaux des différents projets	68
3.3.1 Résultats du projet BRAINBOW	68
3.3.2 Résultats du projet HYRENE	74
3.4 Conclusion et Perspectives	78
4 Vers un protocole universel de communication pour systèmes neuromorphiques	81
4.1 Le protocole AER	84
4.1.1 Introduction	84
4.1.2 Synthèse sur le protocole AER	87
4.2 Particularité du protocole UDP	88
4.2.1 Notre intérêt autour du protocole UDP	89
4.2.2 Paquets et encapsulations des données	89
4.3 Le protocole GIHON	89
4.3.1 Cahier des charges du protocole GIHON	89
4.3.2 Anticipation de l'avenir	90
4.3.3 Architecture sur le FPGA	90
4.3.4 Machine d'état gérant la synchronisation et l'envoi des données	92
4.4 Architectures pour réseaux de neurones informatique et biologique	95
4.4.1 Architecture comprenant un réseau de neurones logiciels	95
4.4.2 Architecture comprenant un réseau de neurones biologiques	95
4.4.3 Synthèse sur les architectures mettant en œuvre le protocole GIHON	96
4.5 Mise en œuvre du protocole GIHON pour deux plateformes neuromorphiques	97
4.6 Conclusion	98
Conclusion & Perspectives	103
Références bibliographiques de l'auteur	109
Références bibliographiques	115

2 Timothée 2 :5, "et l'athlète n'est pas couronné, s'il n'a pas combattu suivant les règles."

CHAPITRE 1

DE L'ANATOMIE DU NEURONE AUX SYSTÈMES NEUROMORPHIQUES

CE chapitre a pour objectif d'apporter les connaissances nécessaires pour lier la biologie et l'élaboration de systèmes neuromorphiques. Comment allier deux mondes si différents ? Les systèmes neuromorphiques ont été fondés sur une volonté d'imiter le comportement biologique des réseaux de neurones, voire du cerveau, dont la puissance de calcul est tout à fait exceptionnel (jusqu'à 10^{19} instructions à la seconde ([Delahaye, 2001](#))). Cependant la complexité des neurones (unités de calculs élémentaires) et des réseaux de neurones bloquent la route vers la totale compréhension du cerveau.

C'est pourquoi notre étude nous amène à faire des rappels de l'anatomie d'un neurone afin d'en comprendre son activité électrique, dans le but de répondre à une question essentielle : Comment peut-on passer d'observations d'un réseau de neurones biologiques à un réseau de neurones électroniques ? La réponse à cette question passe inévitablement par les modèles mathématiques de neurones, lesquels seront détaillés succinctement dans ce chapitre. La quintessence de toute implémentation matérielle réside dans le choix du modèle de neurone utilisé. Après les avoir ainsi présentés, nous montrerons et expliquerons notre choix qui sera utilisé dans le cadre du projet européen BRAINBOW et de l'ANR HYRENE (**H**Ybridation de **R**Eseaux de **N**Eurones pour l'exploration de fonctions de réhabilitation) qui traitent tous deux de l'hybridation (connexion entre l'artificiel et le vivant).

Cette présentation sera à son tour suivie par un état de l'art de l'implémentation de réseau de neurones sur silicium ou plateforme numérique afin de comprendre les limites et les contraintes des implémentations des réseaux de neurones, liées au support matériel, comme notamment la population de neurones embarquées. Quelque soit l'implémentation matérielle, la taille du réseau de neurones abrité reste fixe mais il demeure possible d'unir plusieurs plateformes neuromorphiques (et donc plusieurs réseaux de neurones) afin d'obtenir un réseau de neurones plus grand et/ou de complexifier une fonction.

A l'instar du système nerveux qui est la connexion de plusieurs réseaux de neurones biologiques, les systèmes neuromorphiques électroniques peuvent être connectés entre eux à l'aide d'un protocole de communication. Après une revue des protocoles utilisés au sein de la communauté neuromorphe, nous verrons que cette interaction se limite aux implantations matérielles et introduirons l'ultime problématique de ce travail de thèse proposant un protocole de communication pouvant unir toutes les plateformes neuromorphiques confondues.

1.1 Morphologie et physiologie d'un neurone biologique

Anatomie du neurone

Un neurone est composé de 4 parties différentes remplissant chacune un rôle complémentaire pour le bon fonctionnement du système nerveux, allant de la génération des signaux électriques aux communications entre les cellules (Figure 1.1). Les neurones sont spécialisés dans le traitement des signaux électriques et existent en plusieurs variétés qui diffèrent tant dans la forme, la taille ou même les propriétés électrochimiques ([Cooper, 2011](#)). Nous allons donc détailler succinctement chaque partie afin de parfaire notre compréhension d'un neurone biologique ([Tritsch et al., 1999](#)) et mieux comprendre les modèles mathématiques.

Soma

Le corps cellulaire d'un neurone se nomme aussi soma. Suivant sa position dans le corps, sa forme et sa taille peuvent varier. Le soma abrite le noyau de la cellule et aussi son matériel génétique. C'est dans cette partie de la cellule que les potentiels d'actions sont réceptionnés du

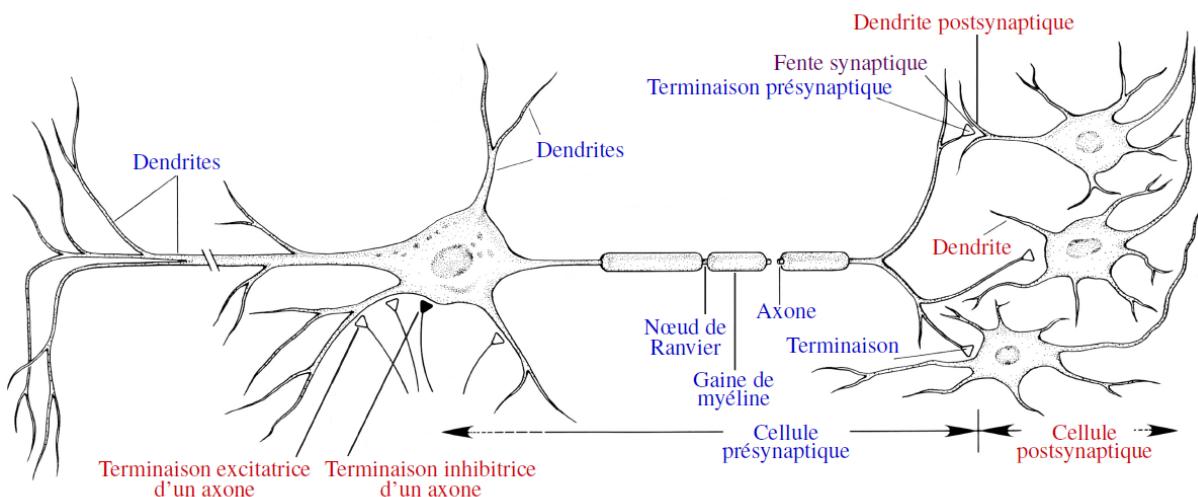


FIGURE 1.1 – Structure d'un neurone. Schéma adapté de (Kandel et al., 2000), mettant en évidence la connexion entre un neurone pré-synaptique et des neurones post-synaptique.

neurone pré-synaptique puis traités, faisant des neurones des unités de traitement d'informations neuronales.

Axone

Cette fine fibre nerveuse (chez l'être humain entre 2 et 18 micromètres de diamètre contre 1 millimètre chez les calamars géants (De Andrés et al., 2010)) pouvant mesurer entre 1 millimètre et près d'1 mètre (chez l'humain (De Andrés et al., 2010)), constitue le prolongement du soma vers ses terminaisons axonales. Le rôle de l'axone est de convoyer les signaux nerveux vers les synapses. Les axones peuvent être recouverts d'une gaine protéique. Cette protéine, appelée myéline, est synthétisée par des cellules gliales dont les cellules de Schwann (pour le système nerveux périphérique) ou les oligodendrocytes (pour le système nerveux central) et permet une plus grande vitesse de passage de l'information nerveuse. Cette gaine est discontinue, séparée par des nœuds de Ranvier placés approximativement à un millimètre d'intervalle (Kandel et al., 2000). Pour finir, l'axone comprend 2 régions : le segment initial où un potentiel d'action prend naissance (plusieurs détails seront donnés dans la prochaine sous-partie) et la partie terminale où les synapses contenant les terminaisons axonales.

Synapses

Les synapses sont les contacts entre 2 cellules et rendent ainsi possible la transmission d'un signal entre un neurone et diverses cellules cibles (neurone, fibre musculaire, glande, etc...). Le neurone se trouvant en amont de l'axone est appelé neurone pré-synaptique et la cellule-cible, en aval de la synapse, est appelé à son tour neurone post-synaptique. Il existe deux catégories de synapses :

- électriques. Elles sont caractérisées principalement par un temps de propagation quasi-nul et par l'absence de période réfractaire (i.e. la synapse est stimulable dès la fin de la transmission). Ces dernières se trouvent en petites quantités dans un corps humain adulte.

- chimiques. Les signaux électriques (ou potentiels d'action) sont réalisés sur la base des courants ioniques qui fluctuent au niveau de la membrane. Les synapses chimiques peuvent transmettre les potentiels d'actions, via des neurotransmetteurs, qui leur permettent d'influer sur l'activité de la (ou des) cellule(s)-cible.

Ces deux catégories de synapses sont réparties dans deux grandes familles de synapses : les synapses excitatrices et les synapses inhibitrices. La première famille vise à dépolariser la membrane du neurone post-synaptique, augmentant ainsi la probabilité de l'apparition d'un potentiel d'action post-synaptique et la seconde a pour but d'hyperpolariser la membrane post-synaptique, provoquant une inhibition ou une diminution d'apparition de potentiels d'actions post-synaptiques. Étant corrélée à des vésicules, l'efficacité d'une synapse fluctue en fonction des circonstances (voir Chapitre 2) et est transcrive par son poids. Derrière ces fluctuations se cache la plasticité synaptique, à l'origine de l'apprentissage et de la mémoire notamment.

Dendrites

Les dendrites sont des prolongements du soma dont elles partagent les organites (à l'exception des lysosomes et du noyau). Ces dendrites se divisent par dichotomies successives. La base trapue des dendrites s'affine vers leur périphérie, ce qui leur donne une apparence arborescente. Les dendrites peuvent être lisses ou couvertes d'excroissances (épines dendritiques), sur lesquelles convergent les afférences synaptiques.

1.2 Physiologie succincte du neurone

La partie la plus intéressante d'un neurone est la membrane plasmique. Cette dernière est à l'origine des potentiels d'action composant n'importe quel message nerveux au sein d'un organisme vivant. En effet, la membrane est capable de varier les différentes concentrations ioniques du milieu intracellulaire et extracellulaire, permettant d'émettre un potentiel d'actions ou une bouffée de potentiel d'action (enchaînement de plusieurs potentiels d'action dans un laps de temps très court). L'étude du comportement de la membrane est la clef de la compréhension de son activité électrique et ultimement la clef donnant accès au modèle mathématique de neurone.

La membrane plasmique

La membrane plasmique neuronale, composée d'une bicouche lipidique traversée par plusieurs types de protéines, forme le contour de la cellule. Elle est une frontière entre 2 milieux liquides : le milieu intracellulaire et extracellulaire. Bien que ces deux milieux possèdent les mêmes cations (ions potassium K^+ , ions calcium Ca^{2+}) et les mêmes anions (ions chlore Cl^-), les concentrations de ces 2 groupes d'ions demeurent différentes de part et d'autre de la membrane. Le milieu extracellulaire est plus riche en Na^+ et Cl^- mais plus pauvre en K^+ que le milieu intracellulaire. Il existe d'autres types d'ions, notamment calcique (Ca^{2+}). Les différences de concentration calcique sont proportionnelles aux variations de potentiels du neurone et sont ainsi utilisées comme marqueur de l'activité électrique neuronale.

La membrane plasmique est composée de canaux ioniques et de pompes ioniques, qui permettront de répondre aux divers stimuli qu'elle subit. Chaque canal ionique est un conduit à sens unique, exclusif à une espèce ionique permettant aux ions de circuler, dont l'ouverture et la fermeture sont actionnées suite à des stimuli spécifiques électriques (pour les canaux tensiodépendants), ou chimiques (pour les canaux ionotropes, sensibles aux messages chimiques

ou neurotransmetteurs), ou mécaniques (pour les canaux mécanosensibles à la pression ou l'étirement). Les pompes ioniques permettent un déplacement actif d'ions. En consommant de l'énergie de la cellule, ces pompes vont permettre simultanément à une espèce ionique d'entrer dans le milieu intracellulaire et à une autre espèce de sortir vers le milieu extracellulaire. Il existe des pompes Na^+-K^+ , qui vont faire entrer les ions K^+ dans la cellule et faire sortir des ions Na^+ . Le principe des pompes ioniques est semblable à celui d'une hélice. Le sens de cette hélice est contrôlé par le potentiel de membrane et peut être réversible dans le cas des pompes $\text{Na}^+-\text{Ca}^{2+}$.

La membrane est donc une frontière perméable, suivant différentes conditions.

Différentes phases de la membrane

Potentiel de repos

Le potentiel de repos est caractérisé par une répartition disparate de charges et d'espèces ioniques entre le milieu extracellulaire et le milieu intracellulaire. Le premier possède un excès d'ions négatifs tandis que le second, au contraire, possède un excès d'ions positifs. Autrement dit, les milieux extracellulaire et intracellulaire sont chargés. A partir de maintenant, la tension de membrane V_m sera la différence entre V_{in} (la tension du milieu intracellulaire) et V_{out} (la tension du milieu extracellulaire).

Le potentiel de repos apparaît lorsqu'aucune stimulation n'intervient sur la cellule et est autour de -65 mV (dépendant de la famille du neurone). Ce potentiel est un point d'équilibre qui se situe entre le potentiel d'équilibre des ions Na^+ (55 mV) et le potentiel d'équilibre des K^+ (-75 mV). Ces deux espèces ioniques sont les actrices principales dans la génération des potentiels d'actions.

Cependant, deux phénomènes sont à l'origine du maintien de ce point d'équilibre. Pour définir l'un des deux phénomènes, nous allons parler de l'équilibre osmotique. Le nombre de particules situées de chaque côté de la membrane doit être le même, quelle que soit la charge de ces particules. Et donc, pour maintenir ce point d'équilibre, la pression osmotique (première force pour le maintien de l'équilibre) des 2 milieux n'est pas égale, alors il se produira un mouvement d'eau suivant le gradient de concentration (c'est-à-dire allant du milieu le moins concentré vers le milieu le plus concentré). Jusqu'à présent, la membrane a été définie comme la frontière entre deux milieux ioniques (contenant cations et anions). Et donc la deuxième force responsable du maintien de l'équilibre est une force électrique qui dépend de la différence de potentiel le long de la membrane. La pression osmotique et la force électrique agissent en permanence pour maintenir le point d'équilibre : le potentiel de repos.

De la genèse d'un potentiel d'action au retour à l'équilibre

Le potentiel de repos est un point d'équilibre stable qui peut être perturbé par les stimulations électriques provenant d'autres neurones. À présent, nous avons toutes les bases nécessaires à la compréhension de la genèse d'un potentiel d'action. Tout débute avec une stimulation qui va rendre le potentiel de la membrane moins négatif que le potentiel de repos. On dira de cette stimulation qu'elle dépolarise la membrane. La tension de membrane va alors quitter le potentiel de repos. Si la stimulation continue, la dépolarisation va permettre à la membrane d'atteindre un seuil de déclenchement (-50 mV), provoquant l'activation ou ouverture des canaux Na^+ . Cette phase correspond à la courbe noire dans la Figure 1.2.

La deuxième phase est représentée par la courbe bleue dans la Figure 1.2. Durant cette phase, plus la dépolarisation augmente et plus les canaux tensiodépendants Na^+ vont s'ouvrir, permettant ainsi aux ions Na^+ de s'infiltrer dans le milieu intracellulaire. Ceci aura pour effet de rendre

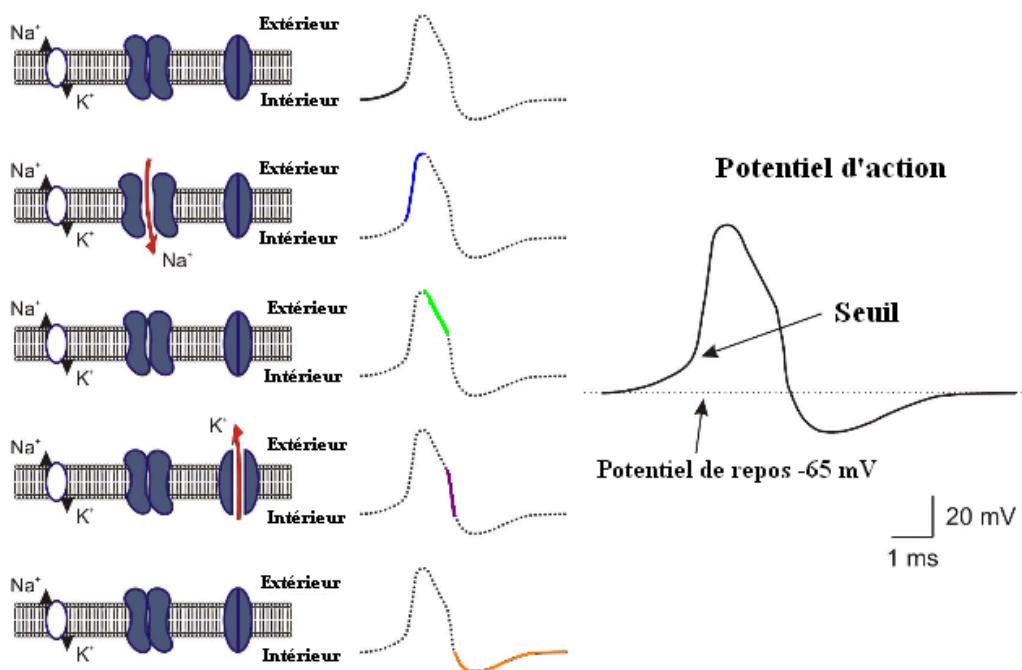


FIGURE 1.2 – Corrélation dans la genèse d'un potentiel d'action entre le potentiel de membrane et les canaux ioniques Na⁺ et K⁺.

le milieu intracellulaire de plus en plus positif et donc d'augmenter encore plus rapidement la dépolarisation. Ce flux ne s'arrêterait qu'à l'atteinte du potentiel d'équilibre de Na⁺ (+55 mV). Cependant, cette valeur ne sera jamais atteinte, car, dans un premier temps, les canaux Na⁺ vont se refermer spontanément (inactivation des canaux Na⁺).

Il s'agit alors de la troisième phase de la genèse d'un potentiel d'action (courbe verte de la Figure 1.2). Dans un second temps, les canaux K⁺ vont s'ouvrir à leur tour (quatrième phase représentée avec la courbe violette dans la Figure 1.2). Ces deux mécanismes repolarisent la membrane. On appellera période réfractaire absolue, la période de décroissance du potentiel d'action. Cela signifie qu'aucun autre potentiel d'action ne pourra être formé à cause de l'inactivation des canaux Na⁺.

L'ouverture des canaux K⁺ va provoquer la fuite des ions K⁺ vers le milieu extracellulaire. Ce flux va faire tendre le potentiel de membrane vers le potentiel d'équilibre du potassium (-75 mV), provoquant une hyperpolarisation (augmentation de la séparation de charge conduisant à rendre le potentiel de membrane plus négatif) de la membrane. Les canaux K⁺ vont ensuite se refermer et le potentiel de la membrane va être ramené vers le potentiel de repos (point d'équilibre de la membrane). L'excédent d'ions est évacué, soit par l'intermédiaire des canaux passifs, soit par l'intervention de la protéine NaK-ATPase. Cette dernière phase (courbe orange dans la Figure 1.2) est appelée période réfractaire relative, car l'hyperpolarisation a éloigné le potentiel de membrane du seuil de déclenchement, rendant la génération d'un autre potentiel d'action plus difficile. Les potentiels d'actions créés par les ions Na⁺ ne durent généralement qu'une milliseconde tandis que les potentiels d'actions à base de Ca²⁺ dure jusqu'à 100 millisecondes.

Propagation d'un potentiel d'action

Une fois le potentiel d'action généré, ce dernier va se propager le long de l'axone à la manière d'une vague. En effet, si un point de la membrane couvrant l'axone est dépolarisé au-delà d'un certain seuil, un potentiel d'action est créé dans cette région en réponse à l'ouverture des canaux Na^+ . Cette dépolarisation locale va, à son tour, provoquer une dépolarisation de la région voisine et va engendrer un potentiel d'action similaire au potentiel d'action précédent. Ce phénomène va se répandre de proche en proche le long de l'axone. A chaque fois qu'une région de la membrane a émis un potentiel d'action, elle passe par une période réfractaire absolue et une période réfractaire relative, ce qui va assurer que le potentiel d'action n'emprunte que la direction menant à la cellule post-synaptique. L'ingéniosité de l'enchaînement des périodes réfractaires implique que si une région de l'axone, jusque là inactive, venait à émettre un potentiel d'action à son tour, ce dernier ne pourra pas être transmis, vu que les régions voisines traversent leurs périodes réfractaires et sont par conséquent inactivables. Ce mécanisme, couplé à la vitesse de propagation d'un potentiel d'action, garantit ainsi l'unicité et l'uni-directionnalité de la transmission d'un potentiel d'action.

D'ailleurs, la vitesse de propagation d'un potentiel d'action est liée au calibre de l'axone qu'il parcourt. La myéline et les noeuds de Ranvier interviennent pour dépasser cette limite. Dans le paragraphe concernant les axones, nous avons rappelé que les axones sont recouverts d'une gaine discontinue ou de segments réguliers de myéline. A la fin de chaque segment, se trouve une fine région appelé « noeuds de Ranvier ». Les noeuds de Ranvier comportent un très grand nombre de canaux tensiodépendants Na^+ , ce qui permet d'intenses dépolarisations en réponse à la dépolarisation parcourant l'axone. De son côté, un segment de myéline permet de diminuer l'effet capacitif de la membrane (réduisant ainsi la perte de charge électrique le long du trajet) et permet d'augmenter l'effet résistif de chaque noeud. La réduction de la capacité de la membrane et l'augmentation de l'effet résistif des noeuds permet aux potentiels d'actions de se déplacer beaucoup plus rapidement de noeuds en noeuds. Cette vitesse peut atteindre jusqu'à 120 m/s. Les noeuds vont aussi booster l'amplitude des potentiels d'action, les empêchant de s'atténuer le long de l'axone. Le potentiel d'action est communément associé à la loi du tout ou rien. D'un point de vue macroscopique, seuls deux cas peuvent se présenter. Premièrement, la variation de la tension de membrane génère un potentiel d'action, auquel cas il sera transmis aux synapses sans atténuation (ceci est le cas «tout»). Deuxièmement, la variation de la tension de membrane n'est pas suffisante pour générer un potentiel d'action, alors il ne se passe rien (ceci est le cas «rien»).

Pour conclure ce paragraphe sur le potentiel d'action et sa propagation, on peut dire que la genèse d'un potentiel d'action est soumise à des forces et des mécanismes physiques et chimiques pouvant être mesurés. Une manière de décrire le comportement d'un neurone serait alors de retranscrire par un système d'équations les mouvements ioniques ou l'évolution du potentiel de la membrane plasmique.

1.3 Principes de modélisation neuromorphique

Un modèle de neurone impulsionnel est une description mathématique de certaines propriétés d'un neurone, créé pour imiter et prédire le comportement biologique d'un neurone. Ces modèles vont décrire les dynamiques régissant les variations de la tension de membrane. Parmi ceux-ci, certains modèles décrivent le potentiel de membrane par une variable unique, en mettant de côté la dimension spatiale, et sont appelés modèle à compartiment simple. Dans ce groupe,

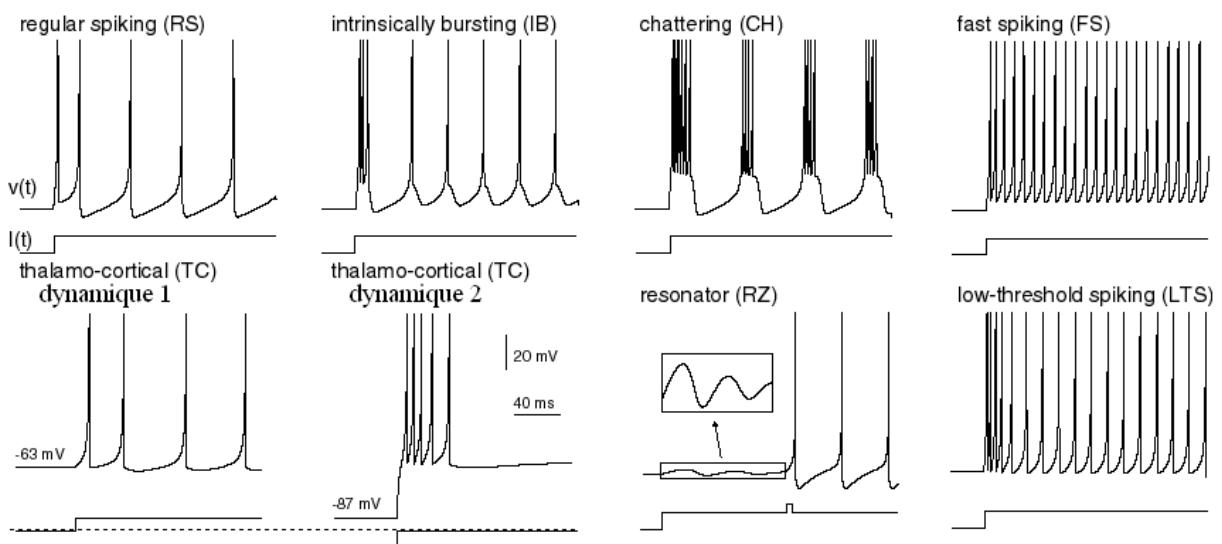


FIGURE 1.3 – Famille de neurones dans (Izhikevich, 2003). Chaque famille est caractérisée par sa réponse $v(t)$ suite à une stimulation en courant $I(t)$.

beaucoup de dynamiques de neurones biologiques peuvent être reproduites grâce à des modèles de neurones reposant sur les conductances ioniques. Parmi ces modèles à conductances, se trouve le modèle Hodgkin-Huxley (HH). Les modèles à compartiment simple permettent en effet de décrire avec précision les dynamiques de neurones et ainsi d'émuler le comportement de plusieurs familles de neurones. Cependant, ces modèles ne permettent pas de retranscrire d'autres réalités biologiques telles que la physiologie des dendrites ou des axones. Ces derniers sont intégrés dans d'autres modèles dits à compartiments multiples ou des modèles reposant sur la théorie des câbles.

Cependant les modèles à compartiment unique sont capables de générer plusieurs familles de neurones. La Figure 1.3 présente quelques types de neurones d'après (Izhikevich, 2003).

Dans cette section, nous présenterons uniquement les modèles de neurones à compartiment unique utilisés durant ces dernières décennies dont le modèle Hodgkin-Huxley implanté dans divers travaux de l'équipe portant sur l'analogique et notre choix d'utiliser un autre modèle en numérique. Pour présenter tous ces modèles, nous commencerons par le modèle le plus proche de la biologie puis le modèle le plus simpliste.

1.3.1 Modèle Hodgkin-Huxley (HH)

Le premier modèle de neurone biologiquement réaliste a été présenté par Hodgkin et Huxley ([Hodgkin and Huxley, 1952](#)). Après avoir réalisé des expériences sur les axones géants de calmar, ces deux scientifiques ont mis en évidence trois courants : courant ionique sodium Na^+ , courant ionique potassium K^+ et un courant de fuite. Suite à cette découverte nobélisée en 1963, ils ont aussi mis en évidence que la perméabilité de la membrane plasmique dépend de la variation de son potentiel. En caractérisant cette tensiodépendance, Hodgkin et Huxley sont parvenus à un modèle mathématique prouvant que cette tensiodépendance est suffisante pour expliquer la genèse des potentiels d'actions. Utilisant une technique appelée « voltage-clamp » (le « voltage-clamp » est une méthode expérimentale utilisée par les électrophysiologistes pour mesurer les courants ioniques au travers de la membrane de cellules excitables, qui consiste à

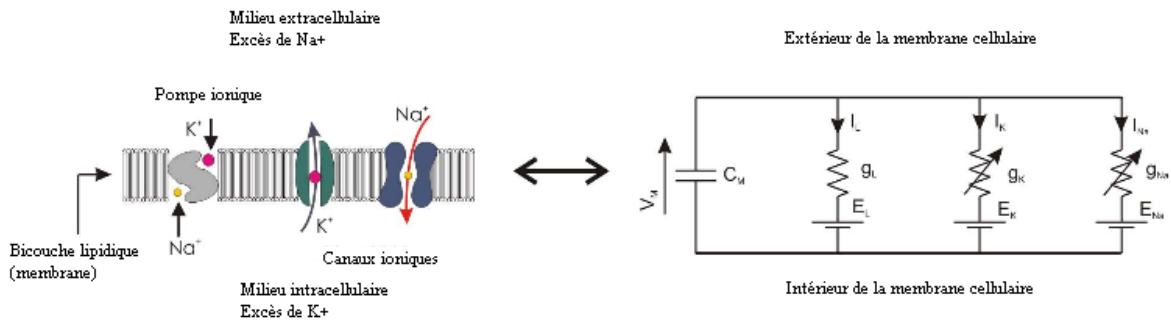


FIGURE 1.4 – Analogie entre les canaux ioniques et les conductances.

changer la tension de la membrane permettant d'activer ainsi les canaux tensiodépendants), ils ont été capables d'enregistrer les différents courants ioniques pour différentes tensions. Ainsi chaque canal ionique a pu être caractérisé par sa résistance (r_{ion}) ou plutôt par sa conductance ($g_{ion} = \frac{1}{r_{ion}}$).

Par convention, nous noterons V_m pour le potentiel de membrane, C_m la capacité de la membrane. Le canal représentant le courant de fuite est représenté par la conductance g_{fuite} . Par analogie, chaque canal ionique est représenté par une conductance. E_{fuite} , E_{K^+} et E_{Na^+} sont les potentiels d'équilibre des ions donnés par les équations de Nernst.

La Figure 1.4 nous montre les courants qui parcourent la membrane et nous permet d'observer :

$$C_m \frac{dV_m}{dt} = -\sum_{ion} I_{ion} = -I_{\text{Na}^+} - I_{K^+} - I_{fuite}. \quad (1.1)$$

Chaque courant ionique (ici noté I_{ion} pour généraliser), issu d'un canal spécifique, dépend de sa conductance g_{ion} , de la tension de membrane V_m et du potentiel d'équilibre ionique E_{ion} , et est décrit de la manière suivante :

$$I_{ion} = g_{ion} \cdot a^p(V_m) \cdot b^q(V_m) \cdot (V_m - E_{ion}). \quad (1.2)$$

Dans cette équation, a et b représentent les probabilités d'ouverture et de fermeture du canal tandis que p et q ne sont que des entiers qui vont varier suivant l'espèce ionique considérée.

Suite à une série de mesures (voltage-clamp), Hodgkin et Huxley ont déterminé la forme de ces fonctions et ont abouti aux équations suivantes :

$$\left\{ \begin{array}{l} C_m \frac{dV_m}{dt} = -g_{fuite} \cdot (V_m - E_{fuite}) - g_{\text{Na}}^- \cdot m^3(V_m) \cdot h(V_m) \cdot (V_m - E_{\text{Na}}) - g_K^- \cdot n^4(V_m) \cdot (V_m - E_K) \\ \frac{dm}{dt} = (1 - m) \cdot \alpha_m(V_m) - m \cdot \beta_m(V_m) \\ \frac{dh}{dt} = (1 - h) \cdot \alpha_h(V_m) - h \cdot \beta_h(V_m) \\ \frac{dn}{dt} = (1 - n) \cdot \alpha_n(V_m) - n \cdot \beta_n(V_m) \end{array} \right. \quad (1.3)$$

avec les potentiels d'équilibre $E_{K^+} = -84$ mV, $E_{\text{Na}^+} = +58$ mV et les fonctions α et β :

$$\left\{ \begin{array}{ll} \alpha_n(V_m) = \frac{0.01 \cdot (V_m - 10)}{\exp(\frac{V_m - 10}{10}) - 1}, & \beta_n(V_m) = 0.125 \cdot \exp(\frac{V_m}{80}), \\ \\ \alpha_m(V_m) = \frac{0.1 \cdot (V_m - 25)}{\exp(\frac{V_m - 25}{10}) - 1}, & \beta_m(V_m) = 4 \cdot \exp(\frac{V_m}{18}), \\ \\ \alpha_h(V_m) = 0.07 \cdot \exp(\frac{V_m}{20}), & \beta_h(V_m) = \frac{1}{\exp(\frac{V_m - 30}{10}) + 1}. \end{array} \right. \quad (1.4)$$

A présent, g_{ion} représente la valeur maximum de la conductance de l'espèce ionique considérée, qui sera pondérée par les éléments sans dimensions n, m et h (probabilité d'ouverture et de fermeture des canaux), variant entre 0 et 1 et représentant respectivement l'activation du canal Na^+ , l'activation du canal K^+ et l'inactivation du canal Na^+ .

Cependant, pour enrichir la dynamique des neurones, il est nécessaire d'ajouter d'autres courants à la définition présentée jusqu'à maintenant. L'adaptation en fréquence est une caractéristique du comportement des neurones de la famille RS (Figure 1.3) qui est obtenue par l'ajout du courant potassium lent. De même, les bouffées de potentiels d'actions sont des caractéristiques des neurones de la famille IB (Figure 1.3) et LTS (Figure 1.3). Pour décrire les neurones IB, il est nécessaire d'ajouter les courants potassium (Yamada et al., 1989), calcium lent (Reuveni et al., 1993) tandis que pour décrire les neurones LTS, les courants potassium lent et calcium faible-seuil (Jahnsen and Llinas, 1984)(Pospischil et al., 2008) vont être ajoutés.

Bien que ces 3 courants additionnels permettent au modèle HH original de décrire de nouvelles dynamiques, ces derniers présentent tout de même des dynamiques non linéaires similaires aux autres courants ioniques.

Le modèle HH est proche de la physiologie d'un neurone grâce à une description exhaustive de l'activation et l'inactivation des diverses canaux ioniques. Mais les fonctions ainsi détaillées montrent qu'il faut posséder une grande puissance de calcul pour utiliser ce modèle. Les équations (1.3) à (1.4) mettent en évidence des comportements sigmoïdiens à la puissance N et des comportements exponentiels (1.4). Ces deux comportements peuvent être rapprochés au comportement d'un transistor bipolaire mais demeurent extrêmement difficiles à réaliser avec un pipeline numérique (Bonabi et al., 2014).

Et au final, malgré la proximité de ce modèle avec la réalité biologique, la complexité computationnelle de ce modèle nous amène à nous poser une question : « Est-il possible de trouver un modèle de neurone plus simple au point de vue computationnelle qui demeure aussi viable biologiquement parlant ? ». Il nous faut donc analyser des modèles plus simples afin de trouver le modèle qui remplira ce compromis.

1.3.2 Modèle "Integrate and fire" (IF) et modèle "Leaky Integrate and Fire" (LIF)

Ce modèle a été l'un des premiers à être proposé. En 1907, Louis Lapicque représente un neurone (Abbott, 1999) par :

$$I(t) = C_M \frac{dV_M}{dt} \quad (1.5)$$

Lorsque V_m atteint une tension seuil V_{th} , un potentiel d'action est généré à la suite duquel V_m retourne au potentiel de repos de la membrane. La fréquence d'émission de potentiel d'action de ce modèle augmente à mesure que le courant augmente. En ajoutant une période réfractaire t_{ref} , le modèle devient plus précis et plus proche du comportement biologique. Ce modèle ne contient pas de mécanisme permettant de mémoriser différents états. Néanmoins, il fût le premier créé dans le but de déchiffrer l'énigmatique activité. Ce modèle ne permet de décrire que 2 familles de neurones : Regular Spiking (RS) et Fast Spiking (FS) (voir la Figure 1.3). Nous avons ainsi vu le modèle le plus complexe reprenant les mécanismes ioniques (le modèle HH) et à l'opposé, le modèle simple IF.

Le modèle LIF (Gerstner, 2002), amélioration du modèle IF, est un circuit basique ayant un condensateur C_m en parallèle avec une résistance R_m , parcouru par un courant $I(t)$. Dans le modèle LIF, le courant est maintenant divisé en 2 parties différentes (Equation 1.6). La première partie est le courant qui charge le condensateur C_m et la seconde partie est le courant de fuite. Le courant de fuite rend ce modèle plus proche du modèle biologique.

$$I(t) - \frac{V_m(t)}{R_m} = C_m \frac{dV_m}{dt} \quad (1.6)$$

1.3.3 Modèle Izhikevich (IZH)

En utilisant la méthode des bifurcations, Izhikevich (Izhikevich, 2003) réduit le modèle HH et propose un modèle composé de 2 équations dans lesquels V_m est la variable d'état représentant la tension de la membrane d'un neurone et u représente la régénération de la membrane, qui prend en compte l'activation des courants ioniques K^+ et l'inactivation des courants ioniques Na^+ .

Lorsque la tension de membrane atteint 30 mV (tension seuil à laquelle on considère le potentiel d'action), V_m et u sont aussitôt remises à des valeurs spécifiques.

$$\begin{cases} \frac{dV_m}{dt} = 0.04 \cdot V_m(t)^2 + 5 \cdot V_m(t) + 140 - u(t) + I(t) \\ \frac{du}{dt} = a \cdot (b \cdot V_m(t) - u(t)) \\ \text{Si } V_m \geq +30mV \quad \text{alors } V_m \leftarrow c, u \leftarrow u + d \end{cases} \quad (1.7)$$

Dans ce modèle, le paramètre a décrit l'échelle de temps de la variable de régénération $u(t)$. Plus la valeur de a sera faible et plus la régénération sera lente.

Le paramètre b décrit la sensibilité de la variable de régénération influant sur les fluctuations sous-seuil du potentiel de la membrane.

Les paramètres c et d sont tous les deux des valeurs de réinitialisation respectivement de la tension de la membrane et de la variable de régénération, après un potentiel d'action. Le paramètre c retranscrit une dynamique provoquée par les conductances K^+ rapides à seuil haut, tandis que le paramètre d retranscrit les dynamiques provoquées par les conductances K^+ et Na^+ lentes à seuil haut. La Figure 1.5 résume le rôle de chaque paramètre du modèle IZH.

Le modèle IZH permet de reproduire le comportement de tous les neurones corticaux connus en ne changeant que 4 paramètres a , b , c et d (voir Figure 1.6). De plus, l'élévation au carré s'implémente facilement grâce aux multiplieurs présents dans les plateformes numériques.

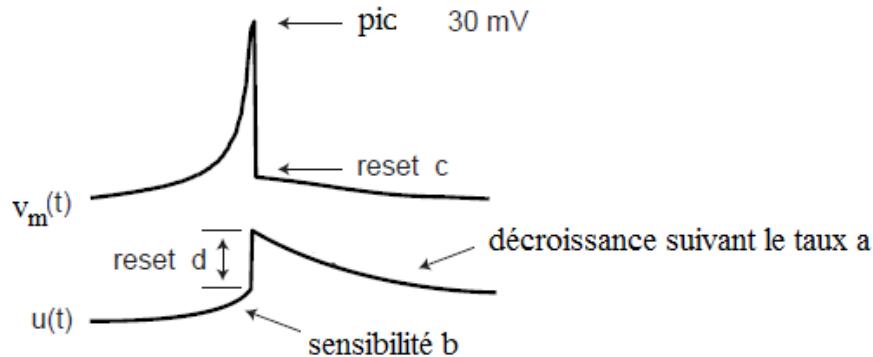


FIGURE 1.5 – Rôle des paramètres a , b , c et d dans l’activité du potentiel de membrane $V_M(t)$ et la régénération de la membrane $u(t)$. Cette Figure est extraite et traduite de (Izhikevich, 2003).

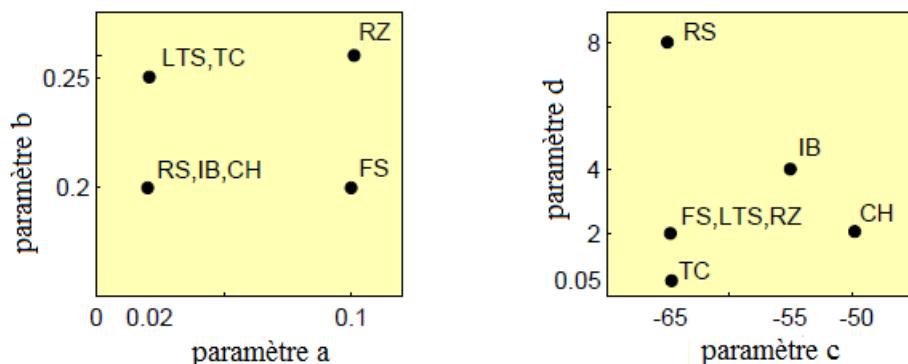


FIGURE 1.6 – Choix des paramètres a , b , c et d en fonction du type de neurone émulé. Les neurones de type RS, IB et CH sont des neurones corticaux excitateurs tandis que les neurones de type FS et LTS sont des neurones corticaux inhibiteurs. Cette figure est extraite et traduite de (Izhikevich, 2003).

1.3.4 Modèle "Adaptive Exponential integrate and fire" (Adex)

Le modèle «Adaptive exponential integrate-and-fire» ou AdEx est aussi un modèle de neurone à deux variables, mis au point par Brette et Gestner (Gerstner and Brette, 2009). La première équation décrit la dynamique du potentiel de membrane et inclut un terme d’activation, exprimé par une exponentielle dépendante de la tension. La tension de membrane est couplée à la seconde qui décrit l’adaptation. Lorsqu’un potentiel d’action est émis, les deux variables sont réinitialisées.

$$\left\{ \begin{array}{l} C_m \cdot \frac{dV_m}{dt} = g_{fuite}(V_m - E_{fuite}) + g_{fuite}\Delta_t \exp\left(\frac{V_m - V_t}{\Delta_t} - w + I\right) \\ \tau_w \cdot \frac{dw}{dt} = a \cdot (V_m - E_{fuite}) - w \\ \text{À } t = t_f \quad V_m \rightarrow V_r, w \rightarrow w + b \end{array} \right. \quad (1.8)$$

Nous retrouvons des similitudes dans ce modèle avec les modèles précédents, à savoir V_m représente le potentiel de membrane. La variable w est la variable d’adaptation, I est la courant entrant sur le neurone, C_m est la capacité de la membrane, g_{fuite} est la conductance des courants de fuite, E_{fuite} le potentiel d’équilibre associé aux courants de fuite, Δ_t le facteur de pente. Pour finir, a est un facteur de couplage de l’adaptation et τ_w est la constante de temps d’adaptation.

Ce modèle permet de reproduire de nombreuses familles présentes dans la Figure 1.3. Le modèle Adex est la combinaison du modèle «exponential integrate and fire» (Fourcaud-Trocme et al., 2003) et du modèle IZH (Izhikevich, 2003). La fonction exponentielle étant difficile à implémenter, nous n'avons pas retenu ce modèle pour notre architecture.

1.3.5 Modèle quartique (Tou)

Le modèle quartique, créé par Jonathan Touboul (Touboul, 2008) sur la base d'analyse de bifurcation mathématique de tous les modèles de la forme $\frac{dv_m}{dt} = F(v_m) - w + I$ avec w comme variable de régénération de la membrane et I un courant. Ce modèle est très proche du modèle IZH, avec cette fois-ci un seuil θ , qui variera en fonction de la famille de neurone représentée. De la même manière que le modèle IZH, la première équation décrit la dynamique du potentiel de membrane. La tension de membrane est couplée à la seconde qui décrit la récupération de la membrane. Lorsque le seuil est atteint, un potentiel d'action est émis et les deux variables sont réinitialisées.

Tout comme le modèle IZH, le modèle Tou possède quatre paramètres sans dimension a , b , V_r et d . L'implémentation de la puissance 4 ajoute un cycle de calcul, ce qui implique une réduction de la vitesse de calcul.

$$\begin{cases} \frac{dV_M}{dt} = V_M(t)^4 + 2 \cdot a \cdot V_M(t) - w(t) + I(t) \\ \frac{dw}{dt} = a \cdot (b \cdot V_M(t) - w(t)) \\ \text{si } v \geq \theta \quad \text{alors } V_M \leftarrow V_R, w \leftarrow w + d \end{cases} \quad (1.9)$$

1.3.6 Le meilleur modèle de neurone pour nos applications

Plusieurs modèles de neurones ont été abordés jusqu'à maintenant, mais notre question reste irrésolue jusqu'à maintenant. Il n'y a pas qu'un seul critère pour déterminer le meilleur modèle de neurone. Ce n'est ni le réalisme biologique, ni la simplicité du modèle, qui va prévaloir, mais plutôt la tâche que l'on souhaite accomplir avec ce modèle. Si notre tâche est d'étudier les très grands réseaux disposant de milliers (ou millions) de neurones, le meilleur choix demeurera le modèle LIF (Cassidy et al., 2011). A contrario, si notre finalité est de pouvoir reproduire l'exact comportement biologique observé suite à des mesures physiologiques, alors le modèle HH est le modèle adéquat.

(Izhikevich, 2004) a classé les modèles de neurones selon deux paramètres : la pertinence biologique du modèle et le coût de son implémentation. Dans notre cas, nous avons deux objectifs, le premier est l'hybridation et le second l'implémentation sur une plateforme numérique. L'hybridation impose de choisir un modèle de neurone qui sur le plan biologique est réaliste, tandis que le second objectif nous impose de ne pas utiliser un modèle de neurone qui nécessiterait beaucoup de ressources pour son implémentation (Figure 1.7). Parmi tous les modèles présentés, le modèle IZH est le plus adéquat dans le cadre de notre travail. Le choix du modèle de neurone est un pas décisif qui nous approche de la problématique de ce travail de thèse. En effet, le modèle IZH est décrit par 2 équations (tenant en 5 additions, 1 soustraction et 4 multiplications), qui seront implémentées en 2 pipelines de calcul. Ce modèle décrit alors un neurone via 4 paramètres statiques et sa dynamique est représentée à l'aide d'un couple de variables d'états (u et v) et du courant I_{izh} .

En résumé, ce modèle offre la possibilité de reproduire une vingtaine de familles différentes de neurones avec 2 pipelines de calculs et 7 paramètres à mémoriser. La Figure 1.7 montre que

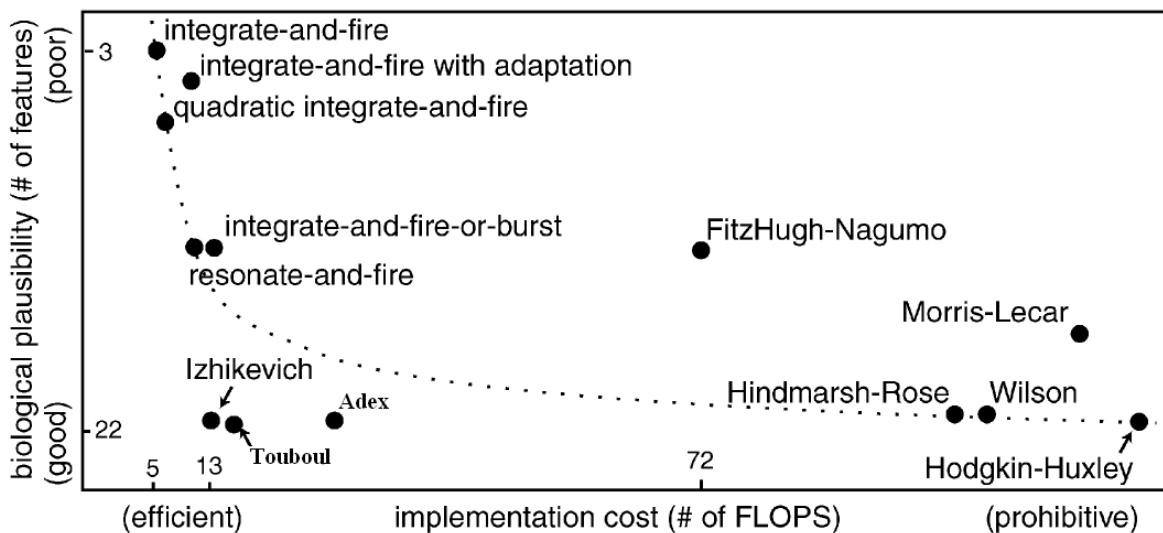


FIGURE 1.7 – Coût de l’implémentation (mesurée en nombre d’opérations) en fonction du réalisme biologique du modèle de neurone (mesuré par le nombre de familles émulables). Cette figure a été copiée de ([Izhikevich, 2004](#)) et mise à jour.

le modèle IZH est l’un des modèles requérant le moins de calculs pour une bonne plausibilité biologique. De plus, comme l’indique ([Izhikevich, 2010](#)), un modèle à base d’équations quadratiques est simple à implémenter dans une architecture numérique. Dans la prochaine section, nous allons parler de la notion de réseau.

1.4 Brève présentation de modèles de synapses

Lors de notre présentation de l’anatomie des neurones et des synapses biologiques, nous avons vu que le rôle des synapses était de convoyer les potentiels d’action d’un neurone pré-synaptique vers des neurones post-synaptiques.

Le modèle HH représente les synapses par une conductance dépendante du temps et de la tension de membrane à l’instar des neurones HH ([Hill et al., 2001](#)). En revanche, certains modèles de neurones (comme le modèle IZH) représentent les synapses par un courant, appelé abusivement poids synaptique, qui est généré en cas d’émission d’un potentiel d’action pré-synaptique. Dans ce cas nous ne sommes dépendants que du temps.

De plus, nous trouvons dans la littérature, des synapses enrichies de propriétés plus complexes. ([Izhikevich, 2003](#)) nous présente une synapse injectant un bruit afin de prendre en compte le comportement stochastique des synapses biologiques tandis que ([Cassidy et al., 2011](#)) présente une synapse avec une décroissance exponentielle du courant. Enfin, d’autres synapses permettent de réaliser l’apprentissage en utilisant la plasticité à long terme ([Rice et al., 2009](#)) ou à court terme (présenté dans ([Izhikevich and Edelman, 2008](#)) et dans le Chapitre 3).

1.5 De l'implémentation de réseaux vers les réseaux de plate-formes

Ce travail de thèse traite 2 aspects des plateformes neuromorphiques : 1)la conception et l'implémentation numérique d'un réseau de neurones, 2)la communication inter-plateforme neuromorphe hétérogène traitée au Chapitre 4

La nature d'une plateforme neuromorphe dépendra de son support ou origine. En effet, une plateforme neuromorphe peut être informatique (comme par exemple, une simulation d'un réseau de neurones sur ordinateur), électronique (cet ensemble regroupe aussi bien les implémentations analogiques que numériques) et biologique (dans le cadre des expériences des projets BRAINBOW et HYRENE, cela est représenté par des cultures de neurones sur MEAs. Dans un cas général, cet ensemble représente le vivant).

1.5.1 Qu'est ce qu'un réseau ?

Dans le corps humain, les neurones sont organisés en structures variées et sont connectés les uns aux autres par le moyen des synapses. Ces synapses peuvent être excitatrices ou inhibitrices. L'impact d'une synapse se traduit par sa conductance ou son poids synaptique.

Par conséquent, à l'instar des réseaux de neurones biologiques, leurs analogues artificiels comprennent des neurones (sous la forme de modèles mathématiques comme ceux présentés dans la Section 1.3) et des synapses, qui sont les composants élémentaires d'un réseau.

L'étude des réseaux de neurones biologiques ont montré que plusieurs dynamiques interviennent dans un réseau de neurones biologiques. Parmi celles-ci se trouve la plasticité synaptique. La plasticité synaptique est une corrélation entre l'activité d'un neurone pré-synaptique et l'activité d'un neurone post-synaptique, changeant en conséquence le poids synaptique. Ainsi, une synapse plastique implique que le poids synaptique peut varier dans le temps. La plasticité synaptique la plus connue est la STDP (Spike Timing Dependant Plasticity) et est aussi considérée comme le phénomène à l'origine de l'apprentissage (Abbott and Nelson, 2000)(Schmidhuber, 2015). Par exemple, dans le cadre de la STDP, si une synapse est utilisée, elle sera renforcée sinon elle sera dépréciée. Au final, la stimulation d'une synapse dépend de sa famille et de son poids sachant que ce dernier peut varier dans le temps suivant si la synapse est plastique ou non. Dans les chapitres 2 et 3, nous parlerons d'un autre type de plasticité STP (Short Term Plasticity), qui est un élément important pour la création des activités oscillatoires dans les réseaux.

Les propriétés des systèmes neuromorphiques ont toujours évolué en s'inspirant des propriétés biologiques observées. Cette affirmation s'applique aussi bien aux modèles de neurones (voir Section 1.3), à la description des synapses et par conséquent également aux réseaux de neurones.

Dans la suite de ce chapitre, nous allons faire un état de l'art des implantations matérielles neuromorphiques au sein de la communauté scientifique.

1.5.2 Etat de l'art des implantations neuromorphiques

Au travers de cet état de l'art, l'auteur veut montrer les domaines d'applications de l'ingénierie neuromorphe depuis la naissance du domaine à nos jours. Au travers de cette chronologie, nous observons la quintessence de la démarche scientifique qui démarre de l'étude du vivant et aboutit à la création de plateformes neuromorphiques.

Domaines d'applications vus par les pionniers

En 1989, l'expression *Neuromorphic engineering* (traduite par *ingénierie neuromorphique*) a été apportée à la communauté scientifique par Mead (Mead, 1989), désignant les multiples champs d'application voués à l'imitation du système nerveux biologique. C'est ainsi que cette expression vient aussi à désigner les systèmes inspirés de réseaux de neurones biologiques.

A partir de 1989, les pionniers du domaine ont étudié les réseaux de neurones biologiques, ont exploré diverses possibilités pour utiliser les réseaux de neurones dans l'ingénierie neuromorphe pour :

- le traitement de données (Mahowald and Douglas, 1991a).
- l'expérience hybride (Jung et al., 2001) (Le Masson et al., 2002) (Sorensen et al., 2004).
- l'étude des réseaux de neurones (Gerstner, 2002).
- l'étude du traitement de données au sein des réseaux de neurones (Liu et al., 2001).

Ce récapitulatif non exhaustif montre la démarche scientifique des pionniers de l'ingénierie neuromorphe qui, s'inspirant de réseaux de neurones biologiques, sont arrivés à créer des systèmes électroniques copiant les propriétés observées. La rétine artificielle est un exemple parfait. L'étude de réseaux de neurones particuliers (comme les cellules rétinienne (Delbrück and Mead, 1991)) conduira à la réalisation de rétine artificielle (Delbrück, 1993)(Mahowald and Douglas, 1991b) mais aussi à la création du protocole AER (Mahowald and Douglas, 1991b) (dont nous parlerons dans le Chapitre 4).

Implémentations matérielles de réseaux de neurones

Pour présenter les différentes implantations matérielles de réseaux de neurones, nous nous appuierons sur la revue des architectures neuromorphiques (Misra and Saha, 2010) et nous compléterons cette étude avec d'autres articles depuis les années 2010. Ceci nous permettra de montrer l'intérêt croissant pour les réseaux de neurones durant ces dernières années, leurs utilités et leurs potentiels dans le traitement de données. En effet, leurs applications s'étendent à la physique à haute énergie, au traitement d'images (reconnaissance d'objets/d'images/de motifs (pattern), segmentation d'images, traitement vidéo), à la robotique, à la reconnaissance acoustique ou olfactive. Parmi les implantations matérielles de réseaux de neurones décrites dans (Misra and Saha, 2010), les implantations matérielles sont divisées en 3 catégories majeures : l'implémentation analogique (à base de puces dédiées), l'implémentation numérique (à base de FPGA, microprocesseurs, microcontrôleurs ou neuropuces), l'implémentation mixte (mélange entre les 2 premières catégories).

Parmi les implantations mixtes se trouvent les implantations numériques assistées par un ordinateur (Ros et al., 2006), ou alors une architecture analogique/numérique (Zou et al., 2006)(Vogelstein et al., 2007). Les implantations analogiques et numériques se distinguent en plusieurs points présentant soit des avantages, soit des inconvénients.

Nous allons présenter dans un premier temps les avantages puis les inconvénients d'une implantation analogique. Nous avons relevé 3 avantages majeurs qui sont :

- la continuité temporelle des variables,
- une densité élevée d'intégration,

- une puissance de calcul élevée. Il est possible de réaliser facilement des opérations mathématiques complexes grâce aux propriétés intrinsèques des composants.

Cependant, l'approche analogique implique aussi un coût de développement très élevé, et de possibles contraintes sur le modèle de neurone utilisé, mais surtout des variations des paramètres entraînant un réglage fastidieux (Grassia et al., 2011).

En utilisant un raisonnement similaire, dans une implémentation numérique, les variables sont elles représentées d'une manière discontinue, non dense. Malgré ces différences avec une implémentation analogique, une implémentation numérique se distingue par un coût de fabrication faible, par un temps de développement faible et par une configurabilité simple.

De plus selon (Misra and Saha, 2010), un réseau de neurone artificiel (RNA) est spécifié en fonction de la topologie du réseau, la fonction d'activation, l'algorithme d'apprentissage, le nombre et le type des entrées/sorties, le nombre de neurones et de connexions synaptiques. Quand il s'agit d'une implémentation matérielle, la technologie employée, la représentation des données (virgule fixe ou virgule flottante), le stockage des données, les bits de précision, la configuration du réseau (programmable ou fixe) et le degré de cascadabilité sont des facteurs à considérer. (Misra and Saha, 2010) nous permet de remarquer que l'utilisation du FPGA pour implémenter des réseaux de neurones a débuté en 1993 (Daalen et al., 1993) et continue jusqu'à nos jours (Delbruck et al., 2014).

Implémentation de réseaux de neurones sur FPGA

(Pocek et al., 2013) nous montre dans son étude que la technologie des FPGA (Field Programmable Gate Array) a été introduite au milieu des années 80 et a connu un succès à partir des années 2000. Dans la littérature, nous trouvons plusieurs implémentations de réseaux de neurones sur FPGA. Ces dernières sont utilisées pour la reconnaissance de patterns (Weeks et al., 2005)(Rice et al., 2009)(Sabarad et al., 2012), la segmentation d'images (Wang and Terman, 1997)(Schoenauer et al., 2002)(B.Girau and C.Torres-Huitzil, 2007), le traitement d'images vidéo (J.Lopez-Garcia et al., 2005), l'analyse vidéo (Krips et al., 2002)(Yang and Paindavoine, 2003). Dans les exemples précédemment cités, les implémentations impliquent des réseaux de neurones impulsionnels (comme ceux présentés dans la Section 1.3).

Au final, le nombre d'implémentations sur plateforme FPGA n'a cessé d'augmenter depuis 1997. En effet (Nazari et al., 2015b) présente les travaux de (Arthur et al., 2012) (implémentation de 256 IF avec un crossbar 1024 x 256 synapses), de (Wang et al., 2013) (implémentation de 4000 neurones et 1.15 millions de synapses), ou alors de (Bonabi et al., 2014) (implémentation de HH simplifié).

Le dernier point de la discussion concerne la taille des réseaux artificiels. En effet, les neurones sont organisés en réseaux de tailles variées. Une comparaison de plateformes neuromorphiques, exprimée par la taille du réseau en fonction de la complexité du modèle utilisé, est montrée dans la Figure 1.7. D'un point de vue électronicien, l'architecture d'une plateforme neuromorphe n'est que le résultat d'un compromis entre le coût computationnel et la complexité du modèle (directement corrélée à la plausibilité biologique).

Cette dernière est une contrainte sur la taille maximale possible du réseau (comme montré dans la Figure 1.7). Pour obtenir de grands réseaux de neurones (populations de 1000 neurones ou plus), il faut :

- soit implémenter un modèle de neurone simple et un modèle de synapse simple. (Cassidy et al., 2011) présente une implémentation sur une plateforme FPGA d'un million de

neurones LIF.

- soit effectuer les calculs du réseau de neurones en temps accéléré comme la plateforme SpiNNaker (Furber et al., 2013) qui comporte une architecture multi-processeur permettant de simuler jusqu'à 18 000 neurones. Nous avons aussi le système BrainScaleS qui est un empilement de modules composés d'un wafer intégrant 448 puces neuromorphiques et d'un système de routage. Chaque module permet d'émuler l'activité de 512 neurones et 115 000 synapses, effectuant les calculs 10^4 fois plus rapidement que le temps biologique (Rast et al., 2013).

Dans notre cas, le temps réel et la plausibilité biologique du modèle de neurone permettent d'hybrider un réseau de neurones vivants et des réseaux de neurones artificiels dans lequel les neurones artificiels sont interconnectés avec des cellules biologiques pour former les réseaux hybrides. Ces contraintes temporelles et de modélisation nous limitent donc dans le nombre maximum de neurones (voir Chapitre 2).

Tendance vers le bioréalisme

L'état de l'art des implémentations des réseaux de neurones montre que celles-ci avaient pour but premier l'étude des réseaux de neurones et de leurs fonctions intrinsèques. Au fur et à mesure du temps, ces implémentations de neurones ont évolué, en partant de modèles simplistes pour aller vers l'incorporation de mécanismes biologiquement réalistes. Cette tendance vers le réalisme biologique a aussi initié une course vers les expérimentations hybrides faisant émerger une volonté d'abolir la frontière entre le vivant et l'artificial. Très tôt dans l'histoire du neuromorphe, nous pouvons observer des implémentations de réseaux de neurones essayant de prendre en compte l'apprentissage.

De récentes recherches ont démontré que les astrocytes interviennent activement dans le traitement des informations et le contrôle des transmissions synaptiques(Kuga et al., 2011)(Papa et al., 2014). Les astrocytes sont les cellules gliales prédominantes dans notre système nerveux et sont émulés via un tout nouveau modèle (modèle présenté dans (Nazari et al., 2015a)(Nazari et al., 2015b) sur plateforme FPGA) et apportant un nouveau niveau de complexité à la notion de réseau de neurones tel que nous le connaissons.

La recherche du réalisme biologique est une facette de l'ingénierie neuromorphe qui dans chaque implémentation est pondérée par la fonction recherchée et les ressources de la plateforme d'accueil.

Vers l'expérimentation hybride

Avec l'émergence de plateformes neuromorphiques temps réel, est apparue la volonté de connecter des réseaux de neurones artificiels avec des réseaux de neurones biologiques (définition de l'expérimentation hybride ou de l'hybridation). Cette volonté conduit à des expériences (Jung et al., 2001)(Le Masson et al., 2002)(Jung, 2011). (Potter et al., 2014) présente différents travaux autour de l'expérimentation hybride en boucle fermée, qui est caractérisée cette fois-ci par l'interconnexion entre des réseaux de neurones biologiques et leurs homologues artificiels.

Dans sa présentation, (Potter et al., 2014) montre les dernières innovations dans le domaine, comme l'expérimentation hybride en boucle fermée utilisant des MEAs composées de nanotubes de carbones ou nanofils (Bareket-Keren and Hanein, 2012)(Robinson et al., 2013), des expériences *in vitro* (Bonifazi et al., 2013)(Tessadori et al., 2012)(Brewer et al., 2013)(Pimashkin et al., 2013), des expériences *in vivo* (Opris et al., 2012)(Nishimura et al., 2013)(Dhingra et al.,

2013) et des essais cliniques (Afshar et al., 2012)(Bryce Beverlin and Netoff, 2012)(Walter et al., 2012)(Fernandez-Vargas et al., 2013)(Walter et al., 2013).

(Bonifazi et al., 2013) (publication liée au projet BRAINBOW) s'inscrit dans cette lignée d'expérience novatrice et s'est illustré par l'étude de la topologie des réseaux de neurones 2D et 3D, la dissection au laser des cultures de neurones et bien sûr l'étude de l'interaction bidirectionnelle entre un réseau de neurones biologiques sur MEA et notre réseau de neurones numériques.

Les recherches portant sur les prothèses neurales (ou neuroprothèses) est un sujet d'actualité (Jung et al., 2015), dans laquelle l'ANR HYRENE (Joucla et al., 2013) participe également.

1.5.3 Les réseaux dans l'équipe AS2N (Bordeaux)

Au sein du groupe Bioélectronique, dans l'équipe AS2N, la teneur principale était constituée des systèmes analogiques ou systèmes majoritairement analogiques. Par le passé, plusieurs puces ont été créées comme les puces Pamina ou Galway (conçues dans le cadre du projet européen FACETS), qui permettent de reproduire le comportement des neurones selon le modèle HH). Mon travail de thèse propose une nouveauté pour l'équipe AS2N, ne portant pas sur un réseau de neurones analogiques mais entièrement numériques. Ce passage de l'analogique vers le numérique est expliqué par un arrière-plan historique, la communauté neuromorphique et aussi par la présence de projets requérant ce savoir-faire. En effet, l'équipe AS2N est impliquée dans 2 grands projets ayant pour finalité l'hybridation vivant/artificiel.

Le projet BRAINBOW (EU FET 2012-2015)

Ce projet est un projet européen en partenariat avec l'IIT (Istituto Italiano di Tecnologia à Gênes), l'UNIGE (Institut of Genova), le TAU (Tel-Aviv University), et le CNRS avec l'équipe Elibio (Bordeaux) et l'équipe AS2N (Bordeaux). Le but du projet Brainbow est de réaliser une preuve de concept pour la prochaine génération de neuroprothèses qui visent à restituer des fonctions perdues au niveau du système nerveux central en organisant une fusion naturelle entre le dispositif artificiel et son équivalent naturel. Ce cadre d'étude sera d'abord appliqué sur des systèmes neuronaux sains puis sur des systèmes atteints (par exemple en cas de lésion). Dans le dernier cas, nous serons en mesure de remplacer une connexion manquante ou un sous-réseau manquant en développant des modèles opératoires et un système électronique neuromorphique VLSI capable d'agir comme le composant biologique manquant afin de restaurer les fonctionnalités du cerveau (par exemple restaurer la fonction entrée-sortie originelle). Le projet BRAINBOW est découpé en 6 tâches :

1. la construction et la caractérisation de modèles innovants du cerveau en analysant les structures 2D/3D des réseaux de neurones cérébraux. En utilisant l'imagerie calcium et l'enregistrement des signaux électriques biologiques, il sera possible d'analyser dans un premier temps des structures simples pour finir sur les structures complexes du cerveau.
2. le développement d'outils d'étude de l'électrophysiologie à grande échelle et de l'analyse des signaux biologiques multicanaux.
3. l'extraction des entrées et des sorties des fonctions du cerveau. Autrement dit, il s'agit de comprendre les dynamiques et les connexions entre différents réseaux.
4. la création d'un simulateur capable de recréer l'activité électrique des neurones composant un réseau.

5. l'implémentation d'un réseau de 100 neurones. Le réseau de neurones utilisera le modèle IZH et sera reconfigurable afin de pouvoir y mettre les paramètres simulées et de reproduire la fonction observée.
6. la réalisation d'une neuroprothèse bi-directionnelle, en réalisant l'expérience (à l'aide d'un prototype) de remplacer un réseau biologique par un réseau de neurones artificiels.

L'équipe AS2N intervient au sein des tâches n° 4, 5 et 6 : la conception du réseau de neurones numérique. Il s'agit de créer un réseau d'au moins 100 neurones utilisant le modèle de neurone IZH. Le choix du modèle de neurone s'est porté sur le modèle IZH car il permet d'émuler toutes les familles de neurones corticales (emplacement proches de la zone d'implantation de notre neuroprothèse). Les synapses doivent être capables de reproduire les dynamiques excitatrices et inhibitrices et la plasticité à court terme.

Le projet HYRENE (ANR 2011-2014)

Ce projet est un partenariat entre le CEA (Grenoble), l'ESIEE (Paris), l'équipe Elibio (Bordeaux) et l'équipe AS2N (Bordeaux). Le but du projet est d'associer la colonne vertébrale d'une souris avec un réseau de neurones artificiels, dans le but de restaurer la fonctionnalité de la colonne vertébrale sectionnée. Ce projet est divisé en 4 tâches :

1. le repérage des racines dans la moelle épinière, à l'origine de la marche chez l'être vivant.
2. la création d'une fourchette adaptée à la stimulation.
3. l'analyse des signaux électriques provenant de la moelle épinière.
4. la création d'un réseau de neurones capable d'émuler l'activité d'un CPG (Central Pattern Generator).

Le cahier des charges ici est de créer un réseau de neurones capable d'avoir une activité électrique similaire à celles observées dans les CPGs.

Les projets BRAINBOW et HYRENE impliquent tous deux l'implémentation d'un réseau de neurones numérique au sein d'une plateforme munie d'un FPGA, appelée MULTIMED. Embarquée sur cette plateforme se trouve une chaîne de calculs et de traitements permettant d'analyser les signaux biologiques provenant des MEAs afin de détecter la présence de potentiels d'action ou de bouffées de potentiels d'action.

MULTIMED peut intégrer un stimulateur multi-fonction (Shiva) ([Kolbl, 2014](#)) pouvant recevoir des ordres numériques pour calibrer une stimulation en courant appropriée. Entre la chaîne de calcul et la stimulation, se situe le réseau de neurones numérique élaboré durant cette thèse. Au vu des blocs de traitement embarqués, ce réseau de neurones doit utiliser le moins de ressources possibles tout en décrivant une grande variété de familles de neurones.

1.6 Conclusion

Nous avons présenté, dans ce chapitre, les notions élémentaires des neurosciences, en partant de la biologie pour arriver au modèle d'Hodgkin-Huxley, qui est le premier modèle à avoir reproduit la finesse physiologique des courants ioniques dans la genèse de potentiels d'action.

Puis nous avons décrit des modèles plus facilement implémentables numériquement tels que le modèle d'Izhikevich, qui est un bon compromis entre le réalisme biologique et le coût d'implémentation (Figure 1.7).

Ce manuscrit présentera dans sa dernière partie les bases pour la création d'un protocole universel permettant de relier différentes plateformes neuromorphiques hétérogènes. En attendant, nous nous proposons de nous intéresser dans le chapitre suivant à l'implémentation du modèle d'Izhikevich dans une plateforme FPGA.

"Je suis de ceux qui pensent que la Science a une grande beauté, répond-elle à ses interlocuteurs. Un savant dans son laboratoire n'est pas seulement un technicien : c'est aussi un enfant placé en face des phénomènes naturels qui l'impressionnent comme un conte de fées. Nous ne devons pas laisser croire que tout progrès scientifique se réduit à des mécanismes, des machines, des engrenages, qui, d'ailleurs, ont aussi leur beauté propre."

E. Curie, *Madame Curie*, p. 465, Edition Gallimard, 1938

CHAPITRE 2

IMPLÉMENTATION DE RÉSEAU DE NEURONES SUR FPGA

CE chapitre décrit l'architecture de notre implémentation de réseau de neurones reconfigurable, ayant pour base le modèle Izhikevich (IZH), sur une plateforme numérique de type "(Field Programmable Gate Array)" (FPGA) et montre comment nous pouvons partir d'une description mathématique d'un neurone à un réseau de neurones sur une plateforme numérique (FPGA). Dans le chapitre 1, nous avons vu les modèles de neurones. A présent nous allons voir comment implémenter le modèle Izhikevich sur un FPGA. Par étape, nous allons devoir expliquer l'implémentation du modèle de neurone et le choix du modèle de synapse dans l'optique de remplir le cahier des charges défini par les projets BRAINBOW et HYRENE. En effet, le projet BRAINBOW exige un réseau de 100 neurones avec une connectivité moyenne de 60 % (i.e avec 60 synapses par neurone en moyenne), avec des synapses dotées de plusieurs dynamiques telles que la plasticité à court terme (Izhikevich, 2003) et le délai axonal. Notre implémentation comprend également une possibilité d'incorporer du bruit dans le courant de stimulation des neurones ainsi que les dynamiques AMPA et GABA des synapses (Ben-Ari et al., 1997). Le projet HYRENE a pour contrainte d'implémenter un réseau de neurones capable de reproduire les comportements des Central Pattern Generator (CPG), caractérisés par leurs activités électriques en bouffées de potentiels d'action (PA) alternées (Hill et al., 2001)(Hill et al., 2002).

Ces 2 projets ont en commun l'élaboration, l'utilisation d'un réseau de neurones numérique et l'implémentation de ce dernier sur une plateforme numérique FPGA (Multimed). La similarité des 2 projets nous conduit à concevoir les principes de base décrivant un réseau de neurones numérique flexible et configurable. La liaison série étant à l'heure actuelle le seul moyen de communiquer avec la plateforme Multimed, nous allons donc utiliser la liaison série pour configurer et donc décrire les réseaux à émuler.

Dans l'explication de notre implémentation, nous allons montrer le principe général et les mécanismes de notre architecture, qui régissent et ordonnent les différents calculs, montrer pourquoi cette architecture est flexible et comment elle permet, avec des changements mineurs, de remplir les 2 cahiers des charges des projets BRAINBOW et HYRENE. Nous terminerons ce chapitre par les performances obtenues avec le système.

2.1 Principes de fonctionnement d'un FPGA

2.1.1 Le FPGA

Apparus au milieu des années 1980, les FPGAs sont des circuits composés de nombreuses cellules logiques élémentaires librement assemblables. Un bloc logique est de manière générale constitué de plusieurs tables de correspondance (LUT ou Look-Up-Table) et de bascules (Flip-Flop). La LUT sert à implémenter des équations logiques ayant généralement 4 à 6 entrées et une sortie.

Elle peut toutefois être considérée comme une petite mémoire, un multiplexeur ou un registre à décalage. En plus de ces blocs logiques, un FPGA contient d'autres ressources précablées comme des mémoires RAMs, des générateurs d'horloge programmables, des entrées/sorties programmables etc.. Ces éléments sont aussi appelés des ressources et sont bien évidemment en nombre limité au sein d'une même plateforme FPGA.

Plusieurs familles de FPGA existent, comme la société Xilinx, un des principaux fabricants de FPGA. En vue de fournir un environnement de travail plus complet, les développeurs de FPGA proposent des kits de développement comprenant un FPGA comme pièce centrale, mais aussi d'autres composants pouvant être utilisés avec ce dernier (convertisseur analogique/numérique,

convertisseur numérique analogique, écran LCD, port série etc...). Afin de pouvoir configurer un FPGA, il est nécessaire d'utiliser un langage de description matérielle ou un outil de saisie graphique. Après synthèse de cette description, on obtient un fichier de configuration pour le FPGA cible. Pour cette thèse, le langage choisi dans cette thèse est le VHDL et les plateformes FPGA utilisées sont les Spartan6 LX150, Spartan6 LX150t et le Virtex5 LX50.

2.1.2 Le VHDL : VHSIC Hardware Description Language

Le langage VHDL fait partie des standards IEEE (Institute of Electrical and Electronics Engineers) depuis 1987, il est largement répandu dans la réalisation de circuits numériques destinés à être implémentés sur FPGA. Le langage VHDL permet la description d'un système matériel, à savoir son comportement, sa structure et ses caractéristiques temporelles. A cela s'ajoute la possibilité, par le biais de ce langage, de non seulement simuler un système mais aussi de le synthétiser, la synthèse consiste à traduire par des logiciels adaptés (ou synthétiseurs) la description VHDL en une série de portes logiques prêtes à être gravées sur silicium. Par conséquent, en fonction de la capacité du FPGA hôte, il est possible de porter notre architecture et ainsi rendre nos travaux entièrement portables et reconfigurables. La suite logicielle utilisée pour la description et la synthèse est l'outil ISE, fourni par le fabricant Xilinx. Cette suite se compose d'outils pour la description, soit par saisie en ligne, soit par saisie schématique, d'outils pour la simulation (Modelsim, Xilinx ISE Simulator etc...), pour la synthèse et l'implémentation vers le composant cible Xilinx.

Le langage VHDL permet un transfert du design de plateformes en plateformes FPGA de même génération ou de génération supérieure.

2.2 Cahiers des charges hérités des projets BRAINBOW et HYRENE

La plateforme MULTIMED (plateforme sur laquelle se trouvera notre réseau de neurones dans le cadre des projets BRAINBOW et HYRENE) abrite un FPGA Spartan 6 LX-150 (avec une horloge 100 MHz) et possède donc différents éléments tels que les portes logiques, des multiplicateurs, des blocs RAMs (mémoire), des convertisseurs analogique-numérique et le port série RS232. Cette plateforme peut être configurée avec l'aide du langage VHDL et des outils de synthèse (chaîne Xilinx).

La plateforme MULTIMED ([Quotb, 2012](#)) est un système conçu pour recevoir les signaux électriques biologiques issus de cultures de neurones posés sur des MEAs (Multi Electrodes Array ou matrices d'électrodes). Ces signaux sont ensuite traités par des fonctions mathématiques comme le traitement d'ondelettes (qui permet de détecter plus facilement les potentiels d'action grâce à un seuil adaptatif), ou encore des fonctions permettant de détecter des bouffées de potentiels d'actions ([Quotb, 2012](#)). Cette plateforme a été voulue comme un outil efficace pour des équipes de biologistes, mais aussi comme une preuve de concept en vue de la création d'un prototype d'une future neuroprothèse. Il permet également de stocker et/ou d'afficher les signaux détectés, les signaux traités et les potentiels d'action générés par notre réseau de neurones, par le biais d'un écran VGA (Video Graphics Array). La Figure 2.1 nous montre l'architecture générale au sein de Multimed.

Donc de base, les outils embarqués sur Multimed utilisent une certaine quantité de ressources. Pour être jugé valide, l'implémentation de notre RNA (Réseau de Neurones Artificiel) ne peut utiliser plus de ressources que celles disponibles et indiquées dans la Table 2.1.

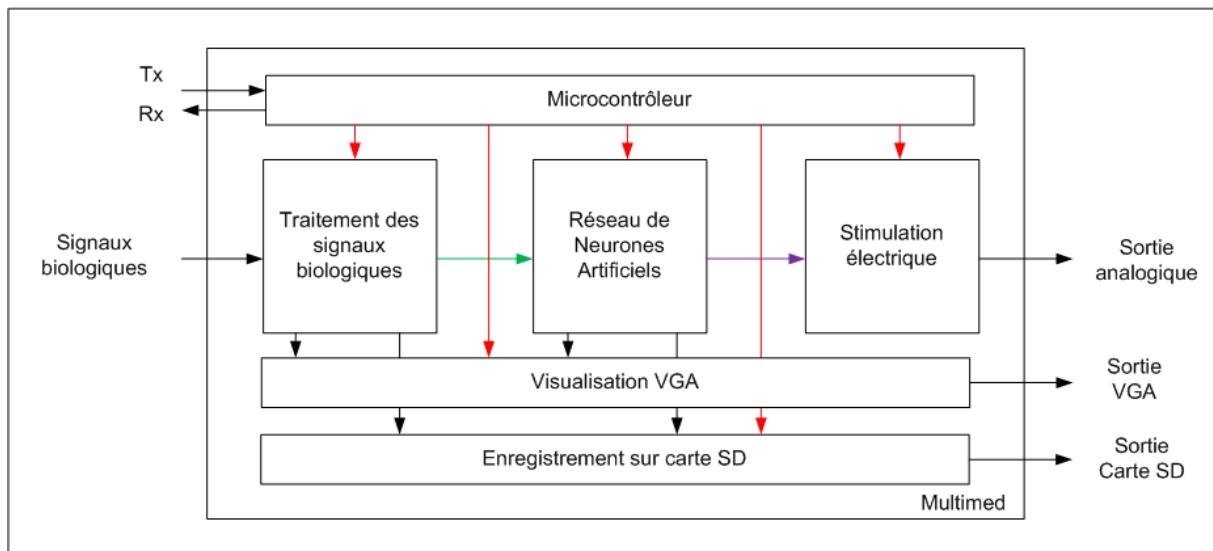


FIGURE 2.1 – Schéma simplifié de l’architecture globale de la plateforme Multimed. L’architecture de Multimed comprend un bloc de traitement de signaux biologiques, duquel sortira un vecteur de potentiel d’action (ou vecteur de PA) représenté par une flèche verte. Un vecteur de PA est la représentation binaire de l’activité d’un groupe de neurones. Ce vecteur de PA ira dans le réseau de neurones artificiel, qui produira un vecteur de PA à son tour, représenté par la flèche violette. La plateforme Multimed a aussi un bloc de communication RS232 dont nous reparlerons plus tard et un bloc de visualisation VGA. Les flèches rouges représentent les directives données par le microcontrôleur.

2.2.1 Cahier des charges BRAINBOW

Dans le projet BRAINBOW, la topologie du réseau de neurone artificiel passe par un ensemble de discussions avec nos partenaires (UNIGE, TAU et l’IIT). A la suite de ces discussions, nous avons déterminé que 100 neurones seront nécessaires pour notre réseau de neurones artificiel. Ce nombre correspond à la population moyenne d’un groupe de neurones (ou d’un cluster de neurones) cultivé sur les électrodes (voir Chapitre 3). Comme nous l’avons vu dans le Chapitre 1, nous avons choisi le modèle de neurone IZH pour son réalisme et aussi la large variété de neurones qu’il peut reproduire. Au vu du but final (interaction entre un réseau de neurones artificiel et un réseau de neurones biologique vivant), il est nécessaire aussi d’avoir des dynamiques au sein de notre réseau de neurones qui soient biologiquement réalistes comme les dynamiques AMPA-ergique et le GABA-ergique (Destexhe et al., 1994), le délai axonal (Katz

Ressources	Totales	Utilisées pour les outils de calculs et d'affichage	Disponibles pour le RNA
Bascules	184 304	12 257	172 047
LUTs	92 152	15 009	77 143
Multiplieurs	180	3	177
RAM 16k	268	126	142
RAM 8k	536	3	533

TABLE 2.1 – Liste des ressources occupées par les blocs de calculs embarqués sur MULTIMED. Par extension, nous avons le nombre de ressources disponibles pour l’implémentation du RNA.

and Miledi, 1965)(Minneci et al., 2012) et la plasticité à court-terme (Izhikevich, 2003). Pour finir, on souhaite pouvoir injecter du bruit sur le courant de stimulation de chaque neurone. Dans ce projet, nous pouvons remarquer que l'architecture des réseaux que l'on souhaite émuler ne peut pas être définie à l'avance. En effet, nous pouvons imaginer avec aisance que pour chaque utilisation de la plateforme MULTIMED, le réseau de neurones artificiels doit pouvoir être adapté. Ceci souligne l'intérêt et la nécessité d'un réseau de neurones artificiels reconfigurable. En résumé le cahier des charges imposé par le projet BRAINBOW est d'intégrer un réseau de neurone qui contient 100 neurones (modèle IZH) avec une connectivité moyenne de 60%. Ce réseau aura des synapses mettant en œuvre la plasticité à court-terme, le délai axonal et la possibilité d'injecter du bruit sur le courant de stimulation des neurones, et sera reconfigurable.

2.2.2 Cahier des charges HYRENE

Dans le projet HYRENE, le réseau de neurones artificiel doit être capable de générer des bouffées de potentiels d'action alternées comme les CPGs (Central Pattern Generator). Le cahier des charges exige que le CPG émulé soit reconfigurable de manière à pouvoir changer la fréquence des bouffées si besoin.

Les CPGs (présentés en détail dans le Chapitre 3) sont des réseaux composés de 2 neurones (pour le plus petit d'entre eux) ou plus. La modélisation de Hill (Hill et al., 2001) nous décrit un réseau de 8 neurones et 12 synapses pour reproduire un CPG biologiquement réaliste.

2.3 Représentation des données

Avant de nous intéresser à l'architecture de notre réseau, nous devons nous intéresser aux nombres de bits alloués à la représentation de chaque paramètre ou donnée mathématique. Les nombres réels dans le monde numérique peuvent être écrits par deux méthodes différentes : la virgule flottante et la virgule fixe. Ces deux méthodes se caractérisent par une stratégie différente dans la représentation des nombres. Dans cette section, nous allons analyser les principes, avantages et inconvénients des 2 méthodes avant de justifier le choix retenu pour notre implémentation.

2.3.1 Principe de la virgule flottante

La virgule flottante consiste à représenter un nombre par un signe s (- ou +1), une mantisse m (aussi appelée significande) et un exposant e (e est un entier relatif généralement borné) et une base b (b sera égal à 10 pour une base décimale ou à 2 pour une base binaire). Par conséquent, un nombre réel r sera alors représenté par $r = s.m.b^e$.

La méthode d'écriture avec une virgule flottante tire son nom du fait qu'il est possible de déplacer la virgule décimale en faisant varier la valeur de l'exposant e . En effet, la mantisse m est représentée par un nombre en base b , généralement de taille fixée, dans laquelle on choisit de placer une virgule à une position fixe : juste avant ou après le premier chiffre ou encore après le dernier chiffre.

Dans tous les cas, pour un nombre donné, la valeur de l'exposant dépend de ce choix (voir Figure 2.2). (Les nombres en virgule flottante peuvent ainsi être vus comme l'équivalent informatique de la notation scientifique).

$$1.987 = 1987 \cdot 10^{-3}$$

nombre réel r = mantisse m . base b exposant e

FIGURE 2.2 – Exemple pour expliquer le principe de la virgule flottante. Le nombre réel 1.987 est représenté via la mantisse 1987 en base 10 et un exposant -3.

2.3.2 Principe de la virgule fixe

En informatique, une représentation d'un nombre en virgule fixe est un type de donnée correspondant à un nombre qui possède (en base deux ou en base dix) un nombre fixe de chiffres après la virgule.

On peut alors concevoir la méthode d'écriture à la virgule fixe comme un cas particulier de la méthode d'écriture à la virgule flottante dans lequel l'exposant e est constant (voir la Figure 2.3).

Les bits à gauche de la virgule représentent, en puissance de 2, la partie entière du nombre, tandis que chaque bit à droite de la virgule, ou « décimale binaire », correspond à l'inverse d'une puissance de 2. Ainsi la première décimale binaire est 1/2, la seconde est 1/4, la troisième est 1/8 et ainsi de suite. Pour un nombre en virgule fixe dans un format de complément à deux, la borne maximale est : $2^{E-1} - \frac{1}{2^{DEC}}$ et la borne minimale est -2^{E-1} où E correspond au nombre de bits de la partie entière signée et DEC au nombre de bits de la partie fractionnaire.

$$1.987 = 0001.111110010110$$

nombre réel r = partie entière E. partie fractionnaire DEC

FIGURE 2.3 – Exemple pour expliquer le principe de la virgule fixe. Le nombre réel 1.987 est représenté sur 25 bits par l'association d'une partie entière (1 bit de signe + 11 bits pour la partie entière) et d'une partie décimale (13 bits). Le nombre ainsi codé est en réalité 1.9871.

2.3.3 Quantification de la correspondance entre la représentation en virgule fixe et la représentation en virgule flottante

Une idée pour quantifier la correspondance entre la représentation en virgule flottante et la représentation en virgule fixe est de prendre 2 populations de neurones composées des différents types de neurones décrits par le modèle IZH. Bien que ces populations contiennent les mêmes types de neurones, la première population sera calculée en virgule flottante (population de référence dans cette expérience) et la seconde sera calculée en virgule fixe. Nous utiliserons le logiciel Matlab qui abrite une bibliothèque permettant d'effectuer des calculs en virgule fixe.

Nous observons alors l'activité de chaque population pendant un temps déterminé (2000 ms) et en les soumettant à un courant I_{stat} .

Les types de neurones ont été décrits par (Izhikevich, 2004). Dans cette étude, nous allons retenir les 4 types de neurones utilisés dans les projets BRAINBOW et HYRENE pour notre comparaison. La dernière question à aborder est : " Comment quantifier la correspondance ?"

Premièrement, nous simulerons deux neurones du même type (RS par exemple), un en virgule flottante et un en virgule fixe. Nous les soumettons alors à un courant I_{stat} variant de 0 à 20. Chaque valeur de stimulation I_{stat} est appliquée pendant 2 000 ms, de manière à obtenir 20 raster plots (représentation événementielle des potentiels d'actions produits par un ou plusieurs neurones) du neurone en virgule fixe et 20 raster plots du neurone en virgule flottante.

Deuxièmement, le raster plot du neurone en virgule flottante est notre référence pour réaliser la comparaison. Pour cela, nous remplissons un vecteur de correspondance suivant une loi simple qui est :

- 1 si un PA en virgule fixe apparaît en même temps qu'un PA en virgule flottant.
- 0 sinon.

Nous sommes ce vecteur de comparaison et divisons par le nombre de PA émis par le neurone en virgule fixe. A la fin, nous obtenons un taux de correspondance entre la virgule flottante et la virgule fixe pour un modèle donné. Ce taux pourra aller de 0 s'il n'y a aucune correspondance à 1 si la correspondance des événements est parfaite. Nous terminons par la moyenne des taux de correspondance sur les 20 raster plots afin d'obtenir un taux de correspondance moyen présenté à la Table 2.2.

Plus le nombre de bits est élevé et plus les calculs en virgule fixe correspondent aux calculs en virgule flottante. Néanmoins dans le projet BRAINBOW, nous avons choisi de représenter nos données sur 25 bits (12 pour la partie entière et 13 pour la partie décimale). Ce choix permet d'utiliser un maximum de mémoire tout en laissant de la marge pour le microcontrôleur embarqué sur la plateforme MULTIMED et aussi de représenter le bruit sans diminuer les performances du système global.

Cependant dans le projet HYRENE, notre architecture utilise une représentation en virgule fixe sur 31 bits avec 19 bits pour la partie décimale. Les paramètres liés à la plasticité à court terme nécessitent des bits supplémentaires de précision pour effectuer les calculs correctement.

Partie Décimale	Regular Spike	Fast Spiking	Chattering	Intrinsically Bursting
6 bits	31.03	17.6	32.21	31.91
7 bits	30.75	17.07	31.94	31.34
8 bits	35.67	27.14	41.45	36.64
9 bits	35.67	27.42	41.96	36.01
10 bits	45.43	31.96	41.61	36.54
11 bits	78.82	86.11	71.35	66.19
12 bits	78.98	86.99	78.13	80.93
13 bits	81.51	91.82	78.45	78.06
14 bits	82.13	87.88	76.72	80.98
15 bits	92.40	92.27	73.57	82.12
16 bits	93.08	94.24	86.94	87.66
17 bits	92.58	94.44	86.68	79.37
18 bits	95.91	96.3	88.54	88.44
19 bits	95.34	96.3	88.19	88.07

TABLE 2.2 – Correspondance (en %) entre la représentation en virgule fixe et la représentation en virgule flottante au travers de plusieurs types de neurones IZH.

2.4 Architecture de notre implémentation

Notre implémentation est constituée d'un module de calcul dédié aux N_n neurones, un module de calcul dédié aux N_s synapses de notre réseau, de mémoire pour stocker les paramètres et les valeurs calculées, d'un module de gestion des calculs et un module de communication RS232 pour pouvoir configurer la mémoire à volonté (Fig 2.4).

Nous allons donc expliquer notre architecture en analysant chacun des blocs qui la compose et dans laquelle chaque neurone et chaque synapse sont identifiés par un ensemble de paramètres individuels (voir Section 2.4.3).

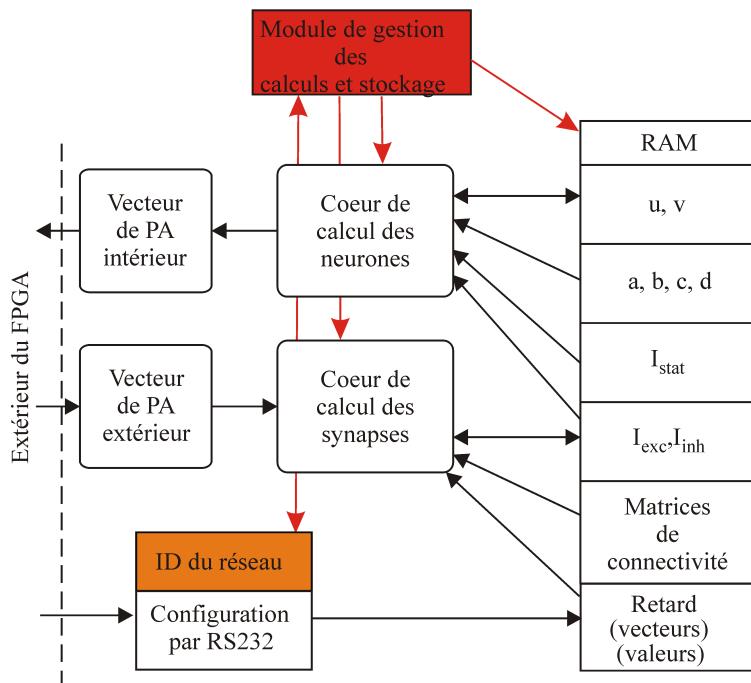


FIGURE 2.4 – Architecture globale de notre implémentation.

2.4.1 Coeur de calcul de neurones

Ce travail de thèse a eu pour point de départ l'étude de (Cassidy and Andreou, 2008). Selon eux, il est possible de prendre le modèle IZH et de créer ce que nous appellerons un cœur de calcul. En effet, l'idée est de créer un module qui a pour tâche de calculer et mettre à jour les valeurs u et v de chaque neurone composant notre réseau. De cette manière, (Cassidy and Andreou, 2008) nous livre une architecture multiplexé temporellement qui va donc calculer les valeurs u et v tour à tour de chaque neurone en utilisant C_n cycles. Cela signifie que cette architecture apporte les coefficients d'Izhikevich (a, b, c et d) et les variables d'état u et v d'un neurone N à un temps donné et le cœur de calcul de neurone donnera le résultat C_n cycles plus tard.

Deux stratégies se révèlent possibles : une architecture multiplexée temporellement et une architecture pipelinée. Dans une architecture multiplexée, mettre à jour les 100 neurones de notre réseau requerra C_n cycles par neurone, pour un total de $100 \times C_n$. L'unique avantage de

cette architecture réside dans son faible coût en ressources mais cette avantage limite le nombre maximal de neurones (Ambroise et al., 2013).

D'un autre côté, une architecture pipelinée (architecture présentée dans ce travail de thèse) permet de faire les mêmes calculs pour les 100 neurones en $(100 + C_n)$ cycles. Intuitivement, faire les calculs en moins de temps nous donne accès à un plus grand nombre de neurones.

A présent revenons au modèle IZH. Celui-ci dépend de 4 paramètres (a , b , c et d) pour reproduire le comportement des neurones corticaux. D'un point de vue mathématique, ce modèle est un système d'équations à 2 dimensions (Izhikevich, 2003) :

$$\begin{cases} \frac{dv}{dt} = 0.04 \cdot v(t)^2 + 5 \cdot v(t) + 140 - u(t) + I(t) \\ \frac{du}{dt} = a \cdot (b \cdot v(t) - u(t)) \\ \text{si } v \geq +30mV \quad \text{alors } v \leftarrow c, u \leftarrow u + d \end{cases} \quad (2.1)$$

Pour rendre le réseau de neurones plus biologiquement réaliste, le courant I_{izh} sera la somme de 3 courants : I_{stat} le courant de polarisation du neurone, I_{inh} un courant qui combine toutes les inhibitions reçues par le neurone et I_{exc} un courant qui combine toutes les excitations reçues par le neurone. (Cassidy and Andreou, 2008) multiplie l'Equation 2.1 par 0.78125 afin de faciliter l'implémentation du modèle IZH en changeant les multiplications en divisions de puissance de 2. En effet, les multiplications (ou divisions) par une puissance de 2 sont en numérique de simples décalages à droite (ou à gauche). Cette simplification nous donne les équations suivantes :

$$\begin{cases} \frac{dV_M}{dt} = \frac{1}{32} \cdot V_M(t)^2 + 4 \cdot V_M(t) + 109.375 - u(t) + I_{stat}(t) + I_{exc}(t) + I_{inh}(t) \\ \frac{du}{dt} = a \cdot (b \cdot V_M(t) - u(t)) \\ \text{si } V_M \geq +30mV \quad \text{alors } V_M \leftarrow c, u \leftarrow u + d \end{cases} \quad (2.2)$$

Nous avons donc implémenté les équations 2.2. Comme le montre la Figure 2.5, nous avons implémenté 2 pipelines, le premier dédié à la mise à jour de la variable v et le second à la mise à jour de la variable u (voir Figure 2.5). En comparaison à l'architecture proposée par (Cassidy and Andreou, 2008), l'ajout de 3 courants au lieu d'un seul allongera le pipeline de calcul de 3 cycles et nous donnera les équations suivantes :

$$\begin{cases} v[n+1] = v[n] + \frac{1}{32} \cdot v[n]^2 + 4 \cdot v[n] + 109.375 - u[n] + I_{stat}[n] + I_{exc}[n] + I_{inh}[n] \\ u[n+1] = u[n] + a \cdot (b \cdot v[n] - u[n]) \\ \text{si } v[n+1] \geq +30mV \quad \text{alors } v[n+1] \leftarrow c, u[n+1] \leftarrow u[n] + d \end{cases} \quad (2.3)$$

Implémentation des dynamiques AMP-ergique et GABA-ergique

La dernière fonction de ce bloc est la décroissance exponentielle des courants synaptiques. Des observations biologiques montrent que les excitations (ou inhibitions) subies par un neurone tendent vers zéro de manière exponentielle (Ben-Ari et al., 1997), suivant une constante de τ_{exc} (ou τ_{inh}). Nous avons décrit cette loi exponentielle par :

$$I_{syn}(t) = I_{max} \cdot e^{\frac{-t}{\tau_{syn}}} \quad (2.4)$$

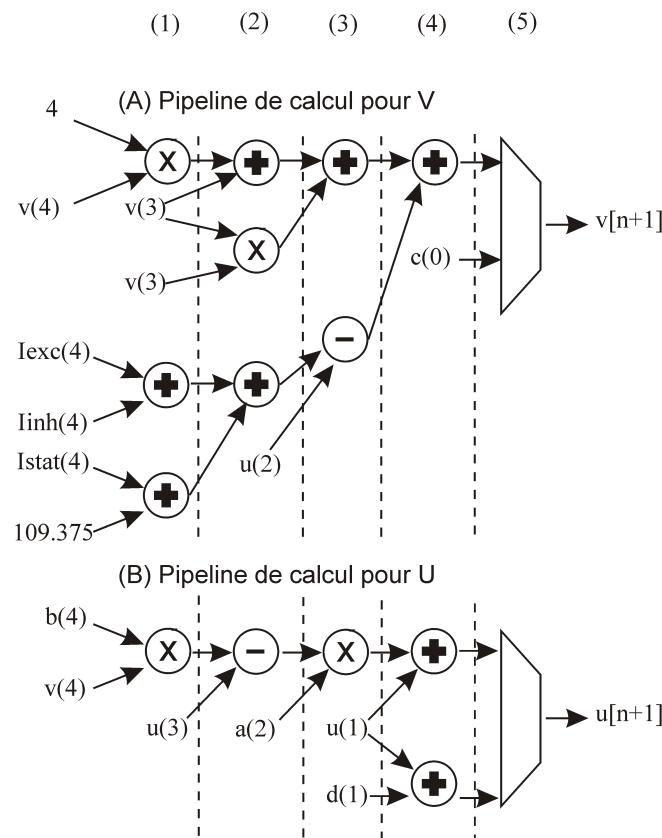


FIGURE 2.5 – Pipeline du cœur de calcul de neurone. Les étapes de calculs (reprenant l'Equation 2.2) ont été mises en évidence par les barres verticales en pointillé. Chacune de ces barres représentent un front montant d'horloge sur lequel le résultat de chaque opération est sauvegardé. Par souci de lisibilité, nous n'avons pas représenté les enchaînements des bascules qui permettent de transférer les opérandes au fil du temps.

Pour alléger notre implémentation, nous avons choisi de calculer la prochaine valeur de la fonction exponentielle, en linéarisant la dérivée de l'Equation 2.4. Nous obtenons ainsi :

$$I'_{syn}(t) = -\frac{1}{\tau_{syn}} \cdot I_{syn}(t) = \frac{I_{syn}(t+T) - I_{syn}(t)}{T}, \quad (2.5)$$

$$\text{soit : } I_{syn}(t+T) = \left(1 - \frac{T}{\tau_{syn}}\right) \cdot I_{syn}(t). \quad (2.6)$$

Avec $T = 1$ ms (pas d'intégration) et τ_{syn} (en ms), l'Equation 2.6 devient alors :

$$I_{syn}(t+T) = \left(1 - \frac{1}{\tau_{syn}}\right) \cdot I_{syn}(t). \quad (2.7)$$

La dernière étape consiste à transposer l'Equation 2.7 pour l'implémentation numérique, ce qui nous donne :

$$I_{syn}[n+1] = I_{syn}[n] - \frac{1}{\tau_{syn}} \cdot I_{syn}[n]. \quad (2.8)$$

Dans notre implémentation, au lieu d'effectuer une division, nous effectuons une multiplication par l'inverse. L'Equation 2.8 nous permet de décrire une décroissance exponentielle si on remplace I_{syn} par I_{exc} (ce courant est toujours positif et est la sommation de toutes les excitations subies par un neurone) ou alors décrit une croissance exponentielle si on remplace I_{syn} par I_{inh} (ce courant est toujours négatif et représente la sommation de toutes les inhibitions subies par un neurone). Dans notre architecture, les valeurs de τ_{syn} sont comprises entre 1 et 2^{DEC} ms (où DEC représente le nombre de bits pour la partie décimale).

Génération du bruit par le processus d'Ornstein-Uhlenbeck

Pour permettre l'activité spontanée et rendre l'activité de notre réseau plus réaliste biologiquement, nous avons implémenté un bruit, en utilisant le processus Ornstein-Uhlenbeck (OU). Ce processus est le meilleur pour modéliser le bruit sur le courant stimulant le neurone (Rudolph and Destexhe, 2005). Ce processus permet d'utiliser le processus de Wiener W_t , pondéré par un degré de volatilité σ , sur une variable x_t et est décrit par l'équation différentielle stochastique suivante :

$$dx_t = \theta(\mu - x_t)dt + \sigma \cdot W_t \quad (2.9)$$

Le paramètre μ est la valeur moyenne, σ est le degré de volatilité causée par des chocs et θ le taux selon lequel les chocs se dissipent et la variable tend vers la valeur moyenne. Le processus OU a été caractérisé avec la condition que les paramètres déterministes σ , μ et θ doivent être supérieurs à 0 et nous confère une variance stationnaire :

$$var(x_t) = \frac{\sigma^2}{2\theta} \quad (2.10)$$

Pour discréteriser l'Equation 2.9, dt sera égal à 1. Nous obtenons ainsi une nouvelle équation implémentée comme le montre la Figure 2.6 :

$$dx_t[n+1] = x_t[n] + \theta(\mu - x_t[n]) + \sigma \cdot W_t[n] \quad (2.11)$$

L'architecture de la Fig 2.6 permet d'utiliser des paramètres individuels pour chaque neurone.

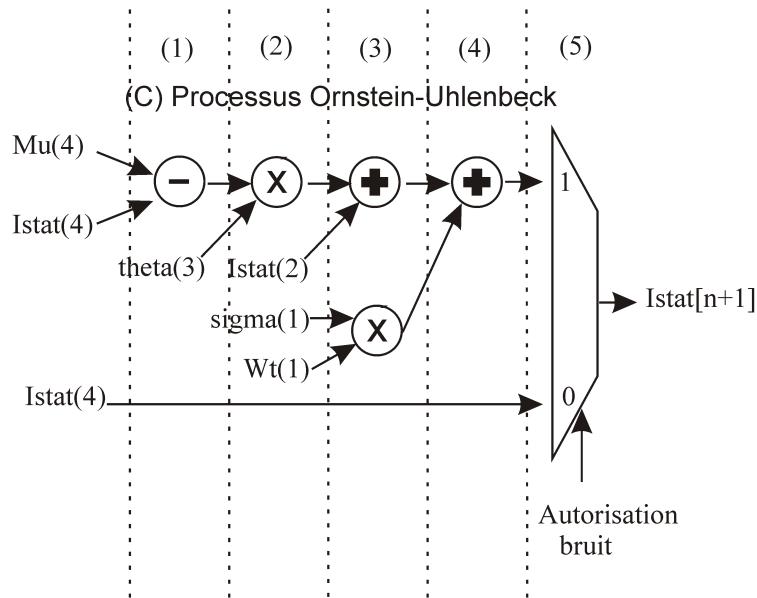


FIGURE 2.6 – Architecture de l’implémentation du bruit. Les paramètres σ , μ et θ sont mis en œuvre conformément à l’Equation 2.11. Ici, W_t représente le processus de Wiener (générateur de bruit). I_{stat} ainsi généré interviendra dans le prochain calcul de l’Equation 2.3.

Dans les études (Touboul, 2004) et (Touboul, 2008), Touboul a par ailleurs testé et validé ce processus de bruit, qui permet de reproduire l’activité spontanée des neurones biologiques. (Touboul, 2008) compare l’activité de neurone stimulé par le bruit et l’activité spontanée. La Figure 2.7 nous montre ainsi cette comparaison et atteste de la similarité des activités du neurone artificiel et du neurone biologique.

Synthèse sur l’architecture du cœur de calcul de neurones

L’aspect temporel est le dernier aspect qu’il nous reste à traiter. En effet, étant donné que le pas d’intégration du modèle IZH est d’une milliseconde, la mise à jour des variables u et v de tous les neurones ainsi que la croissance (ou décroissance) exponentielle doit être effectuée durant une même milliseconde et ce, à chaque milliseconde. Le pas d’intégration de ce modèle représente la première contrainte limitante et définit le nombre maximal de neurones. A la fin de la mise à jour des variables u et v de tous les neurones de notre réseau et de leurs courants synaptiques, nous obtenons alors un vecteur de 100 bits indiquant les neurones qui ont émis un potentiel d’action ou non (1 pour un potentiel d’action émis et 0 dans le cas contraire). Ces bits seront appelés des bits de PA. Ce vecteur de bits permettra d’indiquer aux autres modules (comme le cœur de calcul de synapses) les neurones qui ont émis un potentiel d’action.

Pour finir, le cœur de calcul de neurones prend en entrée les paramètres (coefficients de bruit, coefficients IZH, les courants, variables d’états u et v , l’inverse des constantes de temps des dynamiques GABA-ergique et AMPA-ergique) concernant un neurone et le numéro du neurone. Ces différents paramètres vont traverser une chaîne de bascules qui vont les rendre accessibles au pipeline de calcul à un temps précis. Ce coeur effectue une mise à jour en 6 cycles (voir Figure 2.5).

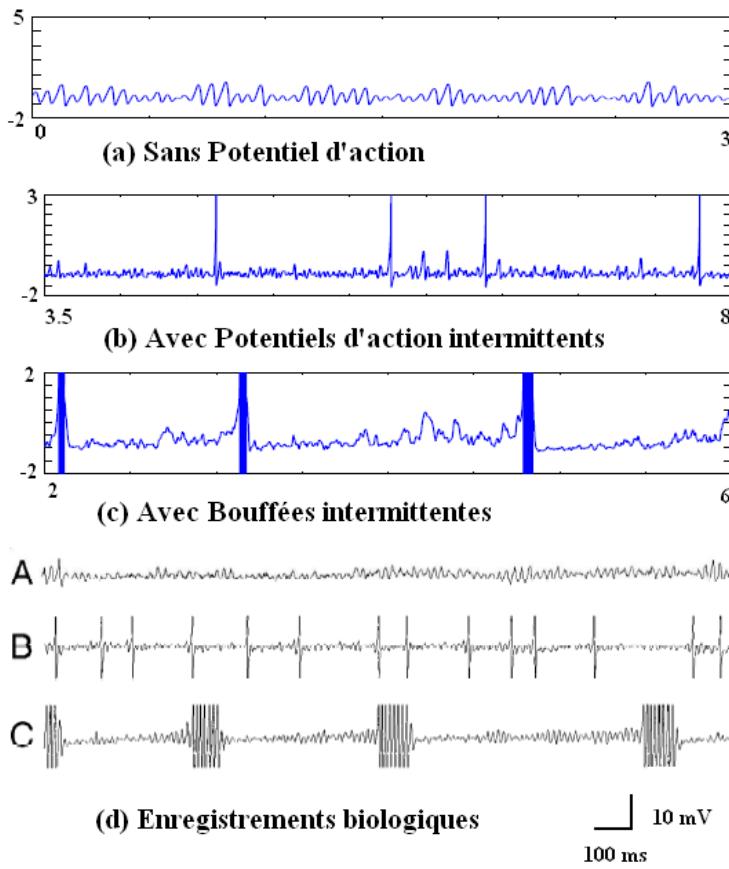


FIGURE 2.7 – Comparaison de diverses activités spontanées d'un neurone artificiel soumis au bruit OU (présentant une activité sans PA, une activité avec PA intermittent et une activité des bouffées de PA intermittentes) et leurs homologues biologiques (Liu et al., 2000). Cette comparaison est extraite de (Touboul, 2008).

2.4.2 Coeur de calcul de synapses

Implémentation de la plasticité à court terme

Utiliser le modèle de synapse proposé dans (Izhikevich and Edelman, 2008) nous permet de reproduire les mécanismes de dépression ou de facilitation d'une synapse selon lequel lorsqu'un neurone pré-synaptique émet un potentiel d'action, le courant synaptique associé au neurone post-synaptique est pondéré par le poids de la synapse liant le neurone pré-synaptique et le neurone post-synaptique. La plasticité à court-terme permet d'ajouter des dynamiques à notre réseau qui le rendront plus réaliste biologiquement parlant et aussi permettra la reproduction de l'activité des CPGs (voir Chapitre 3). Ce modèle est défini par 3 paramètres :

- un facteur scalaire x_{syn} qui va indiquer l'état de la synapse (dépression ou facilitation), qui pondère le poids synaptique et dont la valeur tend toujours vers 1 (voir Table 2.3).
- un pourcentage P qui sera multiplié par le facteur x_{syn} à chaque émission d'un potentiel d'action pré-synaptique. Si ce pourcentage est supérieur à 1, alors cette synapse décrira une

facilitation à court-terme. Dans le cas contraire, si ce pourcentage est inférieur à 1, cette synapse décrira une dépression à court-terme. Dans le cadre d'une facilitation (ou d'une dépression), la valeur ajoutée au courant de stimulation d'un neurone post-synaptique augmentera (ou diminuera) à chaque émission de potentiel d'action.

- une constante de temps t_{syn} de décroissance (ou croissance) exponentiel dans le cas d'une facilitation (ou dépression). Ce modèle s'appelle « plasticité à court-terme » car la facilitation comme la dépression d'une synapse se résorbe lorsqu'aucun potentiel d'action n'a été émis pendant un temps t_{syn} .

Notre cœur de calcul de synapse aura lui aussi 2 comportement différents selon qu'un potentiel d'action pré-synaptique est présent ou non, et donc régi par les équations présentées dans la Table 2.3.

Lorsqu'un potentiel d'action pré-synaptique a été émis	Lorsqu'aucun potentiel d'action pré-synaptique n'a été émis
$I_{syn}[n + 1] = I_{syn}[n] + W_s[n]$	$I_{syn}[n + 1] = I_{syn}[n]$
Avec $W_s[n] = x_{syn}[n].W_{syn}$	
$x_{syn}[n + 1] = P.x_{syn}[n]$	$x_{syn}[n + 1] = x_{syn}[n] + (1 - x_{syn}[n]).\frac{1}{t_{syn}}$

TABLE 2.3 – Système d'équations régissant le cœur de calcul de synapses. Ce système prend en compte 2 cas, soit le neurone pré-synaptique a émis un potentiel d'action soit aucun potentiel d'action n'a été émis.

Implémentation du délai axonal

Le cœur de calcul de synapses gère également le délai axonal. Le délai axonal est le phénomène selon lequel un potentiel d'action émis à un temps t arrive à un temps $t+\Delta t$ au neurone post-synaptique. Dans notre implémentation, le délai axonal est mis en œuvre par un vecteur Délai de 50 bits (registre à décalage) et une valeur D. Nous stockerons le potentiel d'action (1 bit) dans ce vecteur Délai à la position indiquée par la valeur D. Autrement dit, nous pourrons avoir un délai allant de 0 à 49 ms.

Dans le cas sans implémentation du délai axonal, nous avons le comportement suivant. A chaque cycle de calcul, le bit de PA pré-synaptique, les paramètres de la synapse concernée, les courants synaptiques et l'adresse du neurone post-synaptique sont apportés au cœur de calcul synaptique, qui va les utiliser suivant les équations présentées précédemment afin de mettre à jour les valeurs des courants synaptiques en fonction du bit de PA et du poids synaptique. En effet, si le bit de PA est à 1 et que le poids synaptique est négatif, alors seul le courant I_{inh} est mis à jour. Si par contre, le bit de PA est 1 et que le poids synaptique est positif, alors seul le courant I_{exc} sera mis à jour. Les courants mis à jour par le cœur de calcul de synapses sont ensuite stockés à l'adresse indiquée par l'adresse du neurone post-synaptique apportée précédemment.

Dans l'autre cas, avec l'implémentation du délai axonal, le comportement de la synapse est similaire. Cependant, à chaque cycle de calcul, le bit de PA pré-synaptique est stocké dans le registre à décalage de 50 bits (le vecteur Délai) à la D-ième position.

Dans le projet BRAINBOW, ce registre à décalage de 50 bits s'active une fois par milliseconde, permettant ainsi de reproduire un délai axonal de 50 ms au maximum. Ce registre à décalage est une bien meilleure configuration pour représenter le délai axonal d'un point de vue numérique qu'un compteur. Avec l'implémentation du délai synaptique, seul le bit de PA stocké

à la position 0 (i.e le bit de poids faible du registre à décalage Délai) sera pris en compte pour la mise à jour des courants synaptiques.

Le délai axonal permet de décrire une superposition de réseau de neurones en 2D. Plus ces couches sont éloignées et plus le délai est grand. Pour agrandir le délai axonal, il est possible d'envisager en pré-synthèse du code VHDL de changer le temps de décalage.

Synthèse sur l'architecture du cœur de calcul de synapses

Comme nous l'avons défini, le bit de PA indique si un neurone pré-synaptique a émis un potentiel d'action ou non. Cette propriété permet de prendre en compte l'activité du neurone pré-synaptique, mais aussi d'apporter à notre système la possibilité d'exploiter un bit de PA provenant d'un réseau biologique (dans le contexte du projet HYRENE et BRAINBOW). Cela permet bien évidemment d'établir la communication du vivant vers l'artificiel. Nous faisons donc la différence entre la connectivité interne au réseau et la connectivité externe (i.e. la connectivité entre le vecteur de PA venant de l'extérieur vers les neurones du réseau).

Le cœur de calcul des synapses prend en entrée un PA pré-synaptique provenant du vecteur de PA provenant de l'extérieur (ou provenant du réseau) et tous les paramètres relatifs à une synapse (les paramètres de plasticité à court-terme et les paramètres de délai), les courants synaptiques et le numéro du neurone post-synaptique, le numéro de la synapse. En sortie, ce cœur fournit une nouvelle valeur pour un des courants synaptiques, pour le vecteur représentant le délai axonal et pour le paramètre x_{syn} de la synapse.

Ce bloc met 5 cycles pour mettre à jour ces valeurs et fonctionne de manière identique pour la connectivité interne et externe (voir Figure 2.8).

2.4.3 Rôle de la RAM

La mémoire permet de stocker tous les paramètres utilisés par le cœur de calcul de neurones et de synapses, les valeurs calculées par ces derniers.

Concernant les paramètres inhérents au cœur de calcul de neurones et les valeurs issues de ce cœur, nous avons sur N_{bits} ¹ bits :

- N_n (nombre de neurones dans le réseau) paramètres a, b, c et d.
- N_n variables d'état u et v.
- $3 \times N_n$ courants.

Pour le calcul du bruit suivant le processus Ornstein-Uhlenbeck, nous avons les paramètres μ, σ, θ (représentés sur N_{bits} bits) et les bits de permission d'injection. Pour

- $3 \times N_n$ paramètres pour le processus d'Ornstein-Uhlenbeck (μ, σ, θ).
- N_n bits de permission d'injection du bruit.

Le stockage de ces variables et de ces paramètres est intuitif. En effet, ils sont chacun stockés dans un tableau contenant N_n cases. Concernant les paramètres inhérents au cœur de calcul de synapses et les valeurs issues de ce cœur, nous avons :

¹ N_{bits} est la somme du nombre E de bits pour la partie entière et du nombre DEC de bits pour la partie décimale.

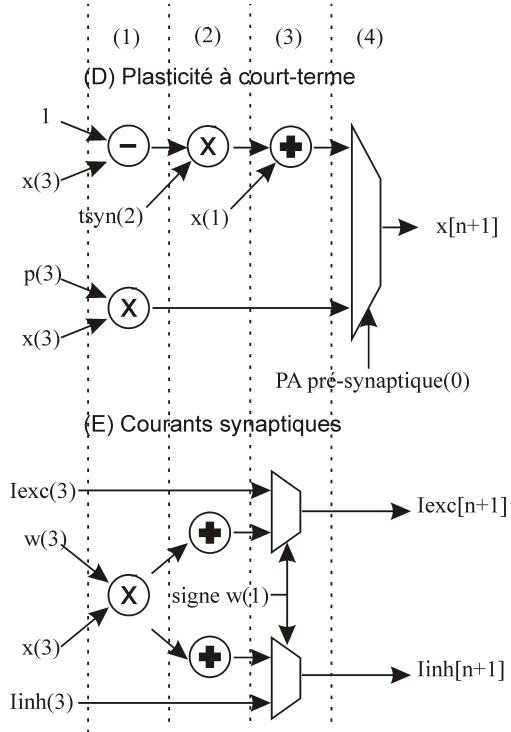


FIGURE 2.8 – Pipeline du cœur de calcul de synapses.

- N_s (nombre de synapses dans le réseau) paramètres w_{syn} , x_{syn} , p et τ_{syn} et adresses de neurones post-synaptiques. Chaque paramètre est représenté sur N_{bits} .
- N_s paramètres pour le délai axonal : D (sur 6 bits) et le vecteur qui lui est associé (sur 50 bits).

L'ensemble des adresses de neurones post-synaptiques est rangé dans la matrice de connectivité (notée m_1 pour la matrice interne et m_2 pour la matrice externe). Chaque adresse de neurone est représentée sur 7 bits et chaque adresse de synapse utilise 13 bits. Nous avons donc 7 matrices indépendantes pour déterminer la destination d'une synapse et ses propriétés. Ces matrices sont dites jumelles car elles possèdent la même taille et la même stratégie de stockage. Deux stratégies de stockage se présentent alors à nous : l'utilisation de N_n tableaux à 60 cases (ce qui confère nos 60 synapses pour chaque neurone) ou alors notre stratégie de listes flexibles. Une liste flexible a pour avantage de récupérer des synapses non utilisées par un autre neurone.

En effet, grâce à cette stratégie, chaque neurone (du neurone 1 à 100) aura une liste d'adresses de neurones post-synaptiques qui se termine par les maillons de fin de chaîne. Les maillons de fin de chaîne sont l'adresse du neurone 0 (repère de fin de liste) et l'adresse du neurone 101 (neurone factice permettant la temporisation au sein des différents coeurs de calcul). Cette flexibilité obtenue a donc un coût de 2 synapses (une aboutissant sur le neurone 0 et une autre aboutissant sur le neurone 101). Un exemple de connectivité est présenté dans la Figure 2.9.

En définitif, la mémoire doit pouvoir stocker 6200 synapses pour les neurones de notre réseau, soit 62 synapses par neurone (60 étant la limite fixée par notre cahier des charges et les 2 restantes permettant le bon fonctionnement du réseau), ainsi que 1200 synapses pour les connexions entre le vecteur de PA extérieur et les neurones de notre réseau (soit suivant le même

principe, (10+2) synapses extérieures par neurone). Le total représentant 7400 synapses.

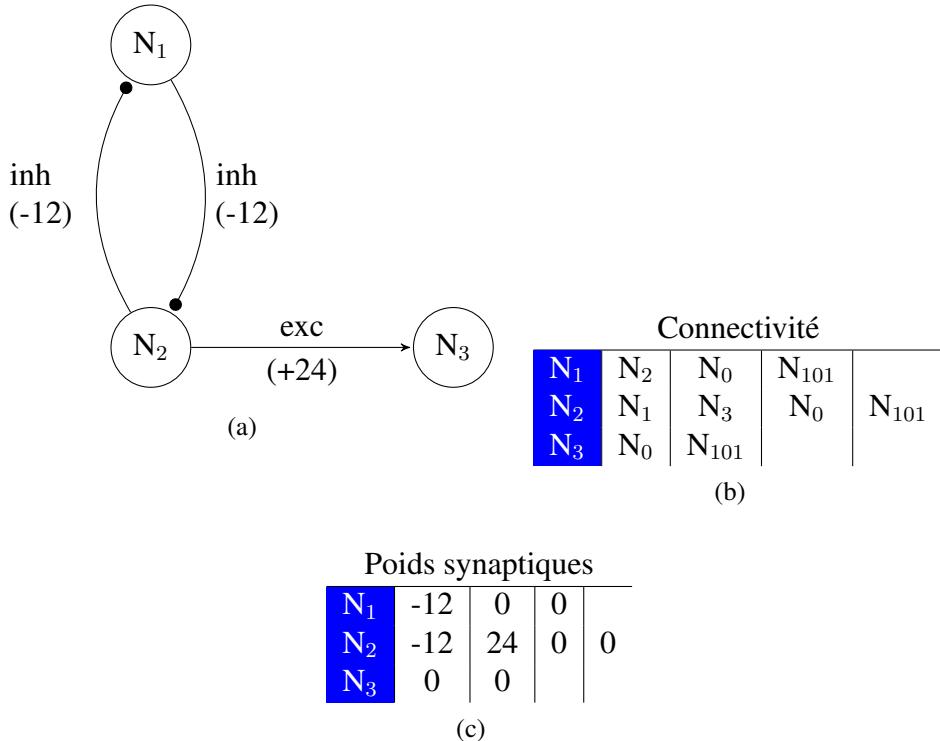


FIGURE 2.9 – Exemple d'un réseau de neurones accompagné de ses matrices de connectivité et de poids synaptiques correspondantes. (a) Dans ce réseau, le neurone N_1 est connecté au neurone N_2 par une synapse inhibitrice (noté inh). Le neurone N_2 est connecté au neurone N_1 par une synapse inhibitrice et au neurone N_3 par une synapse excitatrice (noté exc), tandis que le neurone N_3 n'a pas de neurone post-synaptique. (b) La matrice de connectivité reprend la description faite en (a) et termine chaque liste de neurone post-synaptique par le neurone N_0 et le neurone N_{101} . (c) Les synapses excitatrices (ou inhibitrices) sont caractérisées par des poids synaptiques positifs (ou négatifs). Les matrices de connectivité et de poids synaptiques ont la même taille. Ce principe s'applique également aux matrices de plasticité à court-terme (contenant les x_{syn} , les p et les t_{syn}) et de délai axonal.

2.4.4 Gestion des calculs au sein de notre système

La gestion des calculs dans notre architecture est un élément clé pour la compréhension global de notre système.

L'architecture est régie par 5 états composés de 3 états de calculs, d'un état d'initialisation et un état neutre, représentés dans la Figure 2.10. Le premier est l'état SYN_{EXT} permettant de mettre à jour les courants synaptiques de nos 100 neurones à partir d'un vecteur de PA provenant de l'extérieur, à l'aide du cœur de calcul des synapses. Le deuxième est l'état $NEUR$ dans lequel s'opère la mise à jour des variables d'états u et v de chaque neurone de notre réseau, l'injection du bruit (si elle a été permise) et les dynamiques AMPA-ergique (et GABA-ergique) des courants synaptiques excitateurs (et des courants synaptiques inhibiteurs), à l'aide du cœur de calcul de neurones. Le dernier état de calcul est l'état SYN_{INT} permettant cette fois-ci de mettre à jour les courants synaptiques de nos 100 neurones à partir du vecteur de PA produit par les 100 neurones de notre réseau, à l'aide du cœur de calcul de synapses. Les états SYN_{EXT} et SYN_{INT} ont un fonctionnement similaire, démarrent tous deux à l'adresse de la première synapse et se terminent après l'enregistrement des mises à jour de la dernière synapse.

Ces états sont suivis par un état $IDLE$ qui permet de temporiser dans l'attente de la fin de la milliseconde.

Le dernier état dont nous allons parler est un état d'initialisation nommé INI , dans lequel nous pouvons configurer n'importe quel paramètre de notre réseau.

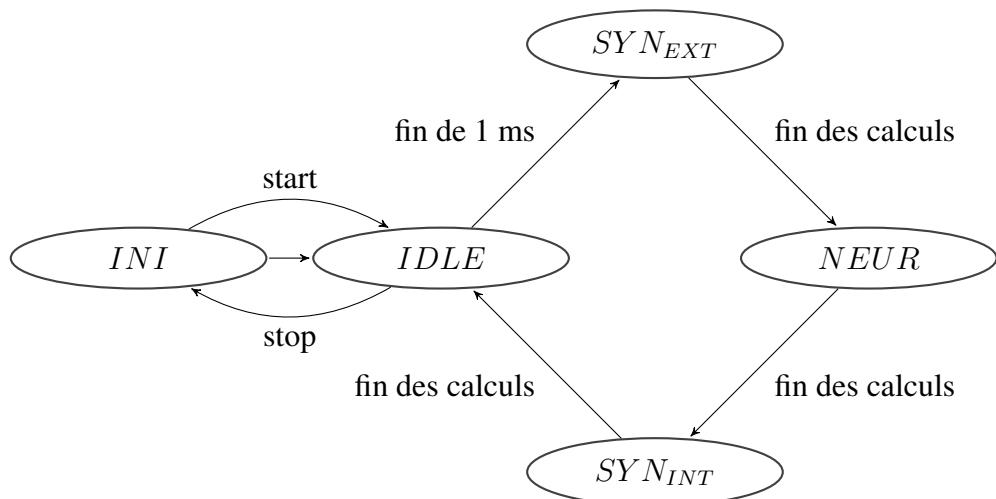


FIGURE 2.10 – Machine d'états de l'architecture montrant la succession des états de calculs (SYN_{EXT} , $NEUR$ et SYN_{INT}), de l'état de configuration offline INI et de l'état de temporisation $IDLE$. Les états de calculs se partagent entre les mises à jour synaptiques (SYN_{EXT} et SYN_{INT}) et les mises à jour neuronales ($NEUR$). Les mises à jour synaptiques s'achèvent lorsque toutes les listes de connectivité ont été parcourues et le pipeline de calcul correspondant est vide. Les mises à jour neuronales s'achèvent lorsque tous les enregistrements des valeurs mises à jours sont terminés.

2.4.5 Reconfigurabilité du réseau via une liaison série

Comme nous l'avons vu dans la section 2.4.3, tous les paramètres du modèle IZH (a , b , c , d , u et v), tous les courants, tous les paramètres des synapses (x_{syn} , p et t_{syn}) et la connectivité interne du réseau et externe au réseau sont stockés dans la RAM. Pour ajouter de la flexibilité

à notre système, nous avons ajouté la possibilité d'utiliser une liaison série (nommée aussi communication RS232 ou connexion UART), présente sur MULTIMED.

Principe de la communication RS232

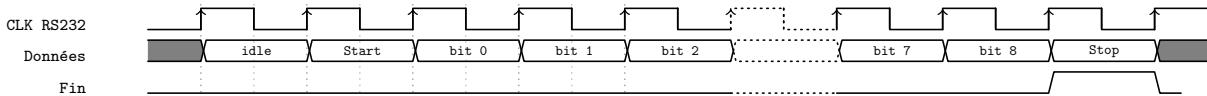


FIGURE 2.11 – Principe de la communication série via un port RS232.

La communication RS232 est une communication série symétrique, utilisant donc un fil pour l'envoi et un fil pour la réception. La Figure 2.11 nous montre que cette opération est encadrée par un bit de start et un bit de stop, qu'il s'agisse de l'envoi ou de la réception des 8 bits de données. Par ailleurs, la communication RS232 permet d'envoyer un caractère en envoyant en série, bit par bit (en partant du bit de poids fort). Du côté de la réception, le caractère envoyé sera reconstitué, bit par bit. A la fin de la réception, un bit nous permettra de confirmer la fin de réception du caractère reçu.

Jeu d'instruction et machine d'état

Pour y parvenir, nous avons élaboré un jeu d'instruction intuitif et donc plus agréable pour les utilisateurs. Pour rendre notre design universel, ce bloc prend en entrée un caractère sur 8 bits et un bit de validité afin de confirmer que ce caractère est celui à prendre en compte. Sur la plateforme MULTIMED, la connexion RS232 est fixée à un débit de 115 200 bits par seconde.

Notre implémentation permet d'identifier un réseau par un nombre entier compris entre 0 et 9. Toutes les instructions créées pour la configuration commencent par l'enchaînement de caractères ‘r’ suivi de l'identifiant du réseau cible. Au sein du bloc chargé de la communication RS232, une machine à état permet de s'assurer un contrôle sur la saisie des paramètres, de sorte qu'uniquement les caractères attendus puissent être pris en compte.

Instructions relatives aux neurones

Les neurones sont définis par les paramètres $a, b, c, d, I_{stat}, I_{exc}, I_{inh}, u, v$ et les 3 paramètres de bruit individuels μ, σ, θ . Pour ce groupe de paramètres, nous allons utiliser la même structure pour les instructions qui permettent de les configurer.

La Table 2.4 nous montre la structure générale des instructions utilisées au sein du système. Les instructions sont délimitées par le caractère initial ‘r’ et le caractère final ‘;’. En utilisant le code couleur dans la Table 2.4, nous avons choisi de démarrer chaque instruction de configuration par la lettre r (en violet) suivie d'un chiffre (en bleu) afin d'identifier le réseau à configurer. Cette précaution a été mise en place dans le cas de l'implémentation de plusieurs réseaux sur le même FPGA et pourrait dans les futures implémentations permettre d'augmenter le nombre de neurones embarqués au sein de la même plateforme (voir Section 2.4.6). Un espace obligatoire suivra le chiffre identificateur du réseau à configurer.

Les 2 caractères suivants vont représenter le paramètre à configurer (en rouge). Si le paramètre ne tient que sur un caractère (comme le paramètre ‘a’ du modèle IZH dans l'exemple A), alors le caractère n°1 sera ‘a’ et le caractère n°2 sera un espace. Par contre, pour les paramètres comme

le courant synaptique I_{stat} (première ligne de l'exemple B), ils seront représentés par le caractère 'i' suivi du caractère 'b'.

Les 3 caractères suivants vont permettre d'indiquer le neurone à configurer (en orange). Du point de vue de la mémoire, ce nombre sur 3 digits décimaux allant de 0 à 100 représente l'adresse du neurone.

Pour finir, toutes les instructions de configuration s'achèvent sur une valeur à affecter au paramètre choisi (en violet). Ces valeurs peuvent être la valeur d'un bit dans le cas de l'autorisation de l'injection du bruit ou être un nombre décimal dans les autres cas, variant entre -99.99999 et 99.99999.

Codage des instructions liées aux neurones

Pour résumer tout en reprenant notre code de couleur, les instructions relatives à un neurone ont été créées sur le principe suivant : **r**, **numéro du réseau à configurer**, suivi d'un espace, **le caractère n°1 du paramètre à configurer**, **caractère n°2 du paramètre à configurer**, **adresse du neurone sur 3 digits**, suivi d'un espace esthétique, **une valeur à affecter (qui peut être signée suivant le cas)**, puis on termine avec **le caractère ';'.**

r	n° réseau	espace	caractère 1	caractère 2	n° neurone	espace	opérande										;
A) Instructions pour configurer les coefficients Izhikevich (a, b, c et d)																	
r	7		a		1	0	0	+ 0 0 .	0 2 0 0 0 0 0 0 ;								
			b					- 6 5 .	0 0 0 0 0 0 0 0 ;								
			c					+ 0 8 .	0 0 0 0 0 0 0 0 ;								
			d														
B) Instructions pour configurer les courants I_{stat} , I_{exc} et I_{inh}																	
r	7		i	b	1	0	0	+ 0 5 .	5 0 0 0 0 0 0 0 ;								
			i	p				+ 0 0 .	0 0 0 0 0 0 0 0 ;								
			i	n				- 0 0 .	0 0 0 0 0 0 0 0 ;								
C) Instructions pour configurer les variables d'états u et v																	
r	7		u		1	0	0	- 1 3 .	0 0 0 0 0 0 0 0 ;								
			v					- 6 5 .	0 0 0 0 0 0 0 0 ;								
D) Instructions pour configurer les paramètres μ , τ et σ																	
r	7		M		1	0	0	0 6 .	0 0 0 0 0 0 0 0 ;								
			T					+ 0 1 .	0 0 0 0 0 0 0 0 ;								
			S					0 0 .	1 0 0 0 0 0 0 0 ;								
E) Instructions pour autoriser/interdire l'injection de bruit																	
r	7		N		1	0	0		1 0 0 0 0 0 0 0 ;								

TABLE 2.4 – Exemple des instructions servant à configurer les paramètres liés aux neurones. Nous avons pris pour exemple ici l'initialisation du neurone 100 du réseau 7. Dans ce tableau, nous avons les instructions dans leurs intégralités telles qu'elles doivent être envoyées en partant de la gauche. Chaque instruction commence par 'r', finit par le caractère ';' et sont de taille variable.

Instructions relatives aux synapses

Les instructions relatives à la configuration d'un paramètre d'une synapse ont été forgées sur un principe similaire aux instructions relatives aux neurones. A la différence que les synapses peuvent provenir de la matrice de connectivité externe (reliant un vecteur de PA provenant d'un réseau de neurone biologique aux neurones du réseau de neurone artificiel) ou de la matrice de connectivité interne (reliant un vecteur de PA fourni par le réseau aux neurones du réseau artificiel).

Les synapses sont définies par les paramètres de la plasticité à court-terme (x_{syn} , p, τ_{syn}), un poids synaptique (w_{syn}) et une valeur de délai D. Pour ce groupe de paramètres, nous allons utiliser la même structure pour les instructions qui permettent de les configurer.

La Table 2.5 nous montre ainsi la structure générale des instructions utilisées au sein du système. Les instructions sont délimitées par le caractère initial 'r' et le caractère final ';' . En utilisant le code couleur dans la Table 2.5, nous avons choisi de démarrer chaque instruction de configuration par la lettre r (en violet) suivi d'un chiffre (en bleu) afin d'identifier le réseau à configurer.

Cette fois, seul le caractère suivant va représenter le paramètre à configurer (en rouge dans la Table 2.5). Par exemple, la lettre 'w' permettra de configurer le poids synaptique w_{syn} .

Le caractère suivant ne peut être qu'un chiffre (1 ou 2) (en jaune dans la Table 2.5). En effet, le chiffre 1 (ou 2) indique que la synapse que l'on souhaite configurer appartient à la matrice de connectivité interne (ou externe).

Nous enchaînons avec le numéro de la synapse (ou aussi adresse de la synapse) à configurer tenant sur 5 digits. Nous continuons par une valeur à configurer (magenta dans la Table 2.5), qui peut être un nombre décimal signé (dans le cas du poids synaptique), ou une valeur décimale positive (pour les instructions configurant x_{syn} , p, τ_{syn} , ou alors l'adresse sur 3 digits du neurone post-synaptique (dans le cas de l'instruction de mapping m) ou bien une valeur de délai sur 2 digits. Nous terminons toutes les instructions par le caractère ';' .

Codage des instructions liées aux synapses

En utilisant le code de couleurs établi, les instructions relatives à une synapse ont été créées sur le principe suivant : **r**, **numéro du réseau à configurer**, suivi d'espace obligatoire, **le caractère du paramètre à configurer**, **un chiffre compris entre 1 et 2**, **adresse du neurone sur 3 digits allant de 0 à 6200**, suivi d'un espace, **une valeur (signée suivant le cas) à affecter**, **le caractère ';' .**

r	n° réseau	espace	caractère 1	caractère 2	n° synapse	espace	opérande	;
A) Instructions pour configurer le poids synaptique w_{syn} d'une synapse								
r	7			W	0 0 0 2	+ 0 9 . 0 0 0 0 0 0 ;		
B) Instructions pour configurer les paramètres liés à la plasticité à court terme (x, p, et t)								
r	7	x	1	0 0 0 2	0 0 0 0 0 0 0 0 0 0 ;			
		p						
		t						
C) Instructions pour configurer le neurone post-synaptique								
r	7	m	1	0 0 0 2	0 0 9 ;			
D) Instructions pour configurer le délai axonal								
r	7	D	1	0 0 0 2	1 0 ;			

TABLE 2.5 – Exemple des instructions servant à configurer les paramètres liés aux synapses. Nous avons pris pour exemple ici l'initialisation de la synapse 2 du réseau 7, qui a pour neurone post-synaptique le neurone 9. Dans ce tableau, nous avons les instructions dans leur intégralité telles qu'elles doivent être envoyées en partant de la gauche. Chaque instruction commence par 'r', finit par ';' et est de taille variable.

Instructions relatives aux effets AMPA-ergique et GABA-ergique

Notre architecture comporte les effets AMPA-ergique et GABA-ergique qui vont permettre de diminuer les excitations ou les inhibitions suivant une constante de temps τ_{AMPA} ou τ_{GABA} . Pour configurer ces constantes de temps, nous utilisons les commandes de la Table 2.6. Ces commandes suivent un schéma identique aux autres instructions en commençant par la lettre 'r' et en s'achevant par le caractère ';'. L'opérande de ces instructions est l'inverse de la constante de temps qu'elles ciblent (en kHz). Par conséquent, si un utilisateur souhaite avoir un constante de temps $\tau_{GABA} = 5\ 000$ ms, il utilisera la commande comme présentée dans la Table 2.6.

r	n° réseau	espace	caractère 1	espace	opérande						;		
A)	Instructions pour configurer la constante τ_{GABA}												
r	7		G		0	0	.	0	0	0	2	0	;
B)	Instructions pour configurer la constante τ_{AMPA}												
r	7		A		0	0	.	1					;

TABLE 2.6 – Exemple des instructions servant à configurer les paramètres liés à l'effet GABA-ergique (A) et l'effet AMPA-ergique. Nous avons pris dans l'exemple (A) l'initialisation de la constante de temps τ_{GABA} du réseau 7, à une valeur de 5000 ms. Dans l'exemple (B), nous avons mis en œuvre l'initialisation de la constante de temps τ_{AMPA} à une valeur de 10 ms. Dans ce tableau, nous avons les instructions dans leur intégralité telles qu'elles doivent être envoyées en partant de la gauche. Chaque instruction commence par 'r', finit par ';' et est de taille variable.

Instruction relative à la configuration

Cette section est spécifique au projet HYRENE, qui ne requiert qu'un petit réseau de neurones composé de 8 neurones. Pour pouvoir l'expliquer, nous allons faire un récapitulatif de l'architecture et sa machine d'état fini.

Notre architecture a adopté un point de vue connexionniste, selon lequel un réseau n'est qu'un ensemble de neurones reliés entre eux par un ensemble de synapses. Précédemment, nous avons vu qu'une synapse est définie par un neurone post-synaptique, un ensemble de paramètres concernant la plasticité et le délai. De plus, la Section 2.4.4 nous a montré que l'état SYN_{INT} (état dans lequel tous les calculs des synapses internes au réseau sont entrepris) démarre par l'adresse de la première synapse du réseau (adresse de départ du réseau) et se termine lorsque la dernière synapse du dernier neurone a été enregistrée.

Parallèlement dans le projet HYRENE, la méthode pour changer la période des bouffées de potentiels d'action est de sélectionner une configuration différente du réseau de neurones (voir Chapitre 3). En effet, une configuration de réseau de neurones sera associée à une période. Seules les synapses diffèrent d'une configuration à l'autre.

Dans l'architecture présentée dans la Section 2.4, nous avons rajouté une case mémoire permettant d'indiquer l'adresse de la première synapse du réseau lors de l'état SYN_{INT} . Ainsi après avoir stocké à la suite les différentes configurations et ajouté la possibilité de changer l'adresse de départ décrivant le réseau, il devient possible de changer de réseau à volonté.

Synthèse sur la configurabilité

Toutes les instructions sont envoyées par la connexion série RS232 et transmises à un bloc qui va les transcrire en parallèle à la RAM au prochain front d'horloge du FPGA. Notre jeu

Instructions pour changer de configuration								
r	7	C	0	0	4	5		;

TABLE 2.7 – Exemple de l'instruction permettant de changer la configuration. Nous avons pris pour exemple ici la configuration du réseau 7. Dans ce tableau, nous avons l'instruction dans son intégralité telle qu'elle doit être envoyée en partant de la gauche. L'instruction indique que la description du réseau démarre par la synapse 45.

d'instructions est complété par une commande pour stopper l'activité du réseau (`r7_;`) et aussi une commande pour le redémarrer (`s ;`). Ces 2 instructions permettent de configurer le réseau offline. Le débit de la communication RS232 sur la plateforme MULTIMED est de 115200 bits par seconde, ce qui signifie qu'un bit est transmis en approximativement 8,68 μ s. La plus courte instruction (le changement de configuration) est transmise et prise en compte en 86,8 μ s tandis que l'instruction la plus longue met 18,2 ms. Notre implémentation peut être configurée de manière offline (en arrêtant le réseau, puis redémarrant ce dernier). En étant offline, le réseau peut recevoir autant d'instructions que nécessaire, tandis qu'en online, nous avons limité à 1 instruction par milliseconde.

2.4.6 Performances du réseau

Les performances du réseau se mesurent au nombre de neurones et de synapses émulables. Ces deux nombres sont définis par une limite temporelle (pas d'intégration du modèle IZH) et surtout la taille de la mémoire présente sur la plateforme FPGA cible.

Premièrement la limite temporelle est en effet le pas d'intégration du modèle IZH. En effet, tous les calculs doivent être effectués au sein de la même milliseconde pour respecter le modèle IZH. Notre implémentation sur une plateforme FPGA Spartan 6 LX 150 a une fréquence critique de 84,8 MHz.

Nous avons vu que les N_n calculs neuronaux seront effectués en $N_n + C_n$ cycles et les stockages sont effectués alors en N_n cycles. De même les calculs synaptiques sont effectués en $N_s + C_s$ et les stockages seront effectués en N_s cycles (C_n et C_s sont respectivement le nombre de cycles pour traverser le cœur de calcul neuronal et synaptique). Ce constat nous donne l'Equation 2.12.

$$(N_n + C_n) + N_n + (N_s + C_s) + N_s \leq C_{total} \quad (2.12)$$

Dans le projet européen BRAINBOW et le projet ANR HYRENE, nous injectons une horloge de 50 MHz. Pour un réseau de N_n neurones et N_s synapses et avec cette horloge, 50 000 cycles sont contenus dans une milliseconde. En injectant $C_n=6$ cycles, $C_s=5$ cycles et $C_{total}=50\ 000$ cycles dans l'Equation 2.12, nous obtenons :

$$2N_n + 2N_s + 11 \leq 50000 \quad (2.13)$$

Le second critère nous permet de déterminer le nombre maximal de neurones et de synapses est la mémoire utilisée pour tous les paramètres. Un neurone est représenté par 4 paramètres d'IZH (chacun sur 25 bits), 2 variables d'état (chacun sur 25 bits), 3 courants (chacun sur 25 bits), 3 paramètres de bruits (chacun sur 25 bits) et un bit de permission du bruit. Au total, nous avons 12 paramètres sur 25 bits et un sur 1 seul bit.

Par ailleurs, une synapse est représentée par 4 paramètres pour la plasticité à court-terme

(chacun sur 25 bits), l'adresse d'un neurone post-synaptique (sur 8 bits²), un vecteur Délai (sur 50 bits) et une valeur de délai axonal (sur 6 bits). Au total, pour une synapse, nous avons 4 paramètres sur 25 bits, 1 sur 8 bits, 1 sur 50 bits et 1 sur 6 bits.

En prenant une plateforme FPGA Spartan 6 LX150, l'ensemble des paramètres pour tous les neurones et toutes les synapses doit tenir dans les 4824 Kb de mémoire (soit 4824 x 10224).

$$N_n(25 \times 12 + 1) + N_s(25 \times 4 + \log_2(N_n) + 6 + 50) \leq 4939776 \quad (2.14)$$

$$\{ 301N_n + (156 + \log_2(N_n)).N_s \leq 4939776 \quad (2.15)$$

Ces contraintes nous donnent donc un système à 2 équations, que nous allons résoudre en choisissant $N_s = N_n \times N_n$.

$$\begin{cases} 2N_n + 2N_s + 11 \leq 50000 \\ 301N_n + 164N_s \leq 4939776 \end{cases} \quad (2.16)$$

Notre architecture nous permet d'avoir un réseau de 155 neurones et 24025 synapses à condition d'utiliser toute la mémoire disponible sur la plateforme FPGA. Pour finir, notre implémentation d'un réseau de 100 neurones et 6200 synapses, pour le projet BRAINBOW, tient dans les ressources disponibles sur Multimed. La Table 2.8 nous montre les ressources nécessaires suivant les options souhaitées.

Architecture en virgule fixe sur								BRAINBOW	HYRENE
Options du SNN		25 bits						31 bits	
délai axonal					X	X	X	X	
Plasticité à court terme			X	X			X	X	X
Bruit		X		X		X		X	
Ressources	Total disponibles	Utilisées suivant l'implémentation							
Bascules	184 304	1619	2130	1877	2388	2161	2414	2161	2672
LUTs	92 152	3112	3742	3543	4166	3699	3884	3699	4301
Multiplieurs	180	28	34	36	42	36	34	36	42
RAM 16 kb	268	8	8	17	17	52	43	52	52
RAM 8 kb	5336	9	12	9	12	9	12	9	12
Total RAM	4824 kb								10

TABLE 2.8 – Liste des ressources occupées par le réseau de neurones artificiel en fonction de la complexité voulue (une implémentation de 100 neurones et 6200 synapses). La colonne bleue représente la configuration choisie pour le projet BRAINBOW (une implémentation de 100 neurones et 6200 synapses) et la colonne violette est la configuration choisie pour le projet HYRENE (10 CPGs avec différentes fréquences, pour une implémentation de 100 neurones et 340 synapses).

²8 bits ont été choisis, en prévision d'un réseau contenant moins de 256 neurones.

2.5 Conclusion

Ce chapitre 2 a permis de montrer que notre implémentation peut émuler un réseau composé de 100 neurones Izhikevich et un total de 6200 synapses, utilisable à la convenance de l'utilisateur.

Les premiers aspects novateurs de cette architecture sont sa modularité et sa reconfigurabilité (offline et online). L'utilisateur peut configurer le réseau qu'il désire et aussi choisir en pré-synthèse les dynamiques qu'il souhaite mettre en action dans son réseau. L'autre aspect novateur de ce travail de thèse réside dans la diversité des dynamiques biologiquement réalisistes (les dynamiques GABA-ergique et AMPA-ergique, la plasticité à court-terme, le délai axonal et l'injection d'un bruit stochastique).

De plus, nous avons exposé des contraintes de notre système, le pas d'intégration du modèle et la mémoire disponible sur la plateforme cible qui restreignent le nombre de neurones maximal. Ce nombre maximal de neurones est intrinsèquement lié à la technologie employée sur le FPGA. En effet, les performances de cette implémentation augmenteront sur des générations de FPGA supérieures au Spartan6 (comme Ultrascale).

Au final, notre implémentation convient ainsi parfaitement au projet BRAINBOW qui requiert 100 neurones et 6200 synapses et peut aussi émuler plusieurs CPGs. Cette architecture peut facilement être utilisée pour le projet HYRENE qui n'utilise qu'un CPG de 8 neurones et 16 synapses.

CHAPITRE 3

MISE EN ŒUVRE DE NOTRE RÉSEAU ET EXPÉRIENCES HYBRIDES

LE troisième chapitre est consacré aux expériences menées dans le cadre des projets BRAINBOW et HYRENE. La flexibilité de notre implémentation permet de configurer notre réseau de neurones à volonté et lui faire émuler différents réseaux biologiques et donc différentes fonctions.

Dans le projet BRAINBOW, le réseau est configuré conformément aux simulations réalisées par une équipe de l'IIT (**Istituto Italiano di Tecnologia** à Gênes). Les configurations visent à imiter et/ou remplacer un cluster d'une centaine de neurones biologiques et mettent à contribution toutes les fonctionnalités décrites dans le chapitre 2.

Dans le projet HYRENE, le réseau est utilisé pour imiter l'activité des "Central Pattern Generator" (CPG). Nous allons présenter un type de CPGs biologiques (celui à l'origine des battements de cœur chez la sangsue (Hill et al., 2001)(Hill et al., 2002)). En effet, l'activité de ce CPG a une activité très proche de celle que nous recherchons dans le projet HYRENE et a servi de première approche à la création des CPGs. En effet, les CPGs décrivent de manière générale une activité électrique rythmique qui sera transmis à un groupe de moto-neurones (neurones qui sont connectés à des fibres musculaires) pour déclencher à leur tour la contraction ou l'extension des muscles. Dans le règne animal, ces activités électriques ont des rapports cycliques et des périodes variables. De même, durant notre étude du CPG biomimétique, nous allons aussi nous concentrer sur une méthode efficace pour changer la période des bouffées de potentiels d'actions (PA).

L'étude des neurosciences évolue chaque jour et requiert de nouveaux outils pour obtenir des connaissances plus précises. Parmi ces outils, nous trouvons les expérimentations hybrides qui consiste à connecter un système artificiel aux cellules biologiques (Le Masson et al., 2002). L'expérimentation hybride, basée sur l'imitation des organismes vivants, peut nous conduire au développement des neuroprothèses ((Hochberg et al., 2006)(Nicolelis and Lebedev, 2009)(Hochberg et al., 2012)). Plusieurs projets internationaux travaillent sur le développement de neuroprothèses, comme le projet européen BRAINBOW (Brainbow, 2012) et l'ANR HYRENE (Joucla et al., 2013), dans lequel ce travail de thèse est impliqué. Ces 2 projets ont un but commun mais diffèrent sur la zone d'implantation de la neuroprothèse et les fonctions à réaliser *in situ*. En effet, la neuroprothèse dans le projet BRAINBOW est un système neuromorphique visant à reproduire l'organisation fonctionnelle des zones endommagées du cerveau. En contraste, la neuroprothèse dans l'ANR HYRENE se focalise sur la microstimulation de la moelle épinière afin de restaurer la locomotion chez les patients paraplégiques.

Pour ces deux projets, un protocole expérimental identique a été utilisé, divisé en trois grandes étapes :

- L'étape I consiste à connecter le réseau de neurones artificiels vers le réseau de neurones biologiques.
- L'étape II consiste à connecter le réseau de neurones biologiques vers le réseau de neurones artificiels.
- L'étape III consiste à interconnecter le réseau de neurones biologiques et le réseau de neurones artificiels.

Dans un premier temps, ce chapitre mettra en évidence au travers de diverses expériences les possibilités de notre réseau en utilisant la plasticité à court-terme, le délai synaptique ou le bruit. Dans un deuxième temps, nous ferons une étude des CPGs avant de terminer sur les deux projets dans lesquels notre réseau intervient.

3.1 Utilisation du réseau de neurones

Le réseau de neurones présenté au Chapitre 2 est ainsi capable d'endosser n'importe quelle description. Nous allons présenter dans cette section, des réseaux de neurones mettant en valeur les différentes options (plasticité à court-terme, délai synaptique et le bruit synaptique) à notre disposition, en prenant tout d'abord un réseau très simple et finissant avec les CPGs. Pour chacun des cas, nous montrerons l'activité des réseaux.

Pour représenter les réseaux de neurones, nous allons adopter une convention, selon laquelle les synapses inhibitrices seront représentées par des segments se terminant par un disque noir. Les synapses excitatrices seront représentées par des flèches.

3.1.1 Réseau "Chenillard"

En utilisant 8 neurones excitateurs RS et les connectant l'un à l'autre par un même type de synapse excitatrice, nous obtenons un réseau en boucle que nous appellerons "chenillard". Le premier neurone, le neurone N_1 , est connecté au neurone N_2 qui sera à son tour connecté au neurone suivant. Cette suite continue jusqu'au neurone N_8 qui sera connecté au neurone N_1 .

En condition initiale, aucun neurone n'est capable d'émettre un PA et seul le neurone N_1 a la valeur de son potentiel de membrane proche du seuil. La Figure 3.1 nous montre l'activité de ce petit réseau (dont les paramètres sont présentés dans la Table 3.1), caractérisée par un enchaînement de PA provoqué par le PA du neurone pré-synaptique (En condition initial, le neurone N_1 émettra un PA lors de la prochaine milliseconde). Le réseau "chenillard" n'a pour seul intérêt que de tester la fonctionnalité du réseau de neurones artificiels.

Paramètres du réseau "chenillard"				
I_{stat}	W	p	τ_{syn}	τ_{AMPA}
5.7	9	1	1 ms	10 ms

TABLE 3.1 – Description du réseau de 8 neurones décrivant le réseau "chenillard". Les neurones utilisés sont des neurones RS. En condition initiale, les neurones N_2 à N_8 ont $v=-65$ et $u=-13$. Le neurone N_1 a $v=20$ et $u=-13$.

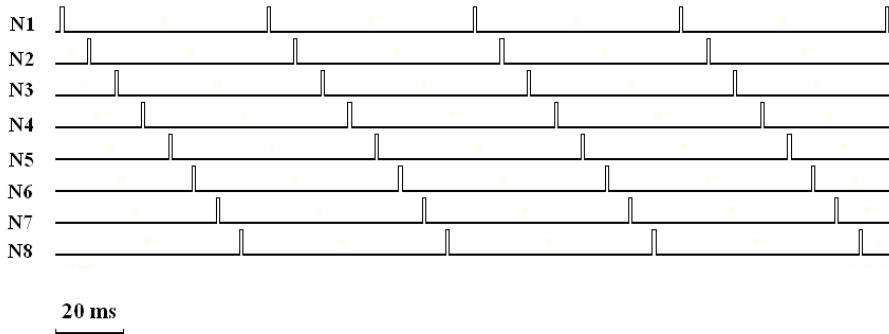


FIGURE 3.1 – Raster plot de l'activité d'un réseau "chenillard". En condition initiale, nous avons imposé au premier neurone d'émettre un PA. Ainsi l'activité des 8 neurones s'enchaîne par un lien de causalité évident. Le PA de N_1 entraîne un PA chez le neurone N_2 .

3.1.2 Réseau "Chenillard" plastique

La Table 3.2 nous montre les paramètres d'un réseau "chenillard" qui cette fois possède des synapses plastiques. La plasticité à court-terme présenté par Izhikevich nous montre que pour décrire des synapses non plastiques, il suffit de mettre les paramètres p et t_{syn} égaux à 1.

En comparant la Table 3.1 (réseau "chenillard") et la Table 3.2 (réseau "chenillard" plastique), nous pouvons remarquer que N_1 a un courant I_{bias} suffisamment élevé pour émettre des potentiels d'actions (PA). La plasticité dans ce réseau va avoir 2 actions :

- en mettant le paramètre $p=0$, la synapse sera dépréciée au maximum à l'émission d'un PA. La synapse ne pourra être réactivée qu'après un certain laps de temps, pendant lequel l'excitation est invalidée.
- en mettant le paramètre $t_{syn}=0.001$, ce paramètre va conditionner le temps de repos nécessaire avant que la synapse soit active de nouveau.

Paramètres du réseau "chenillard" plastique					
$I_{stat}(N_1)$	$I_{stat}(N_2)$ à $I_{stat}(N_8)$	w	p	τ_{syn}	τ_{AMPA}
10	5.7	9	0	1000 ms	10 ms

TABLE 3.2 – Description du réseau de 8 neurones décrivant le réseau "chenillard". Les neurones utilisés sont des neurones RS. En condition initiale, les neurones N_2 à N_8 ont $v=-65$ et $u=-13$. Le neurone N_1 a $v=20$ et $u=-13$.

La Figure 3.2 nous montre l'activité du réseau "chenillard" plastique dans laquelle nous pouvons voir que la plasticité ainsi calibrée empêche l'enchaînement de PAs observés dans la Figure 3.1.

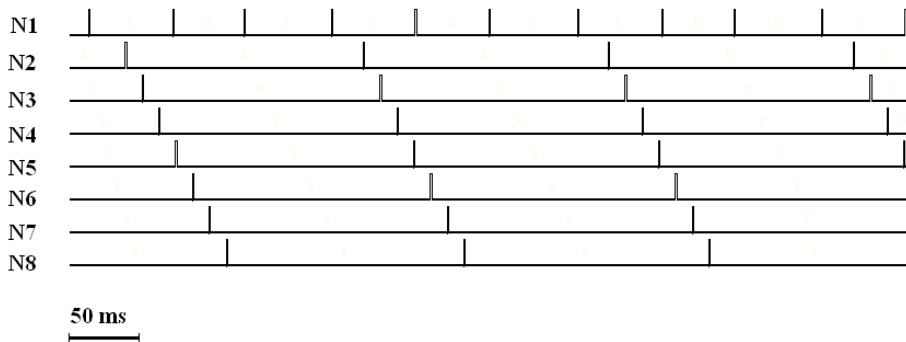


FIGURE 3.2 – Raster plot de l'activité d'un réseau "chenillard" plastique. En condition initiale, nous avons imposé au neurone N_1 un courant $I_{stat}=10$. Ainsi l'activité des 8 neurones s'enchaîne par un lien de causalité évident nuancé par des synapses plastiques.

3.1.3 Réseau "Reconnaissance de motifs temporels"

Le délai synaptique permet de décrire temporellement la transmission et l'arrivée d'un PA et ce phénomène peut être mis en valeur par la reconnaissance de motifs temporels. La Figure 3.3 décrit un réseau de 4 neurones (N_1 à N_4). Le neurone N_1 ne peut émettre un PA uniquement

si les stimulations des 3 neurones excitateurs (N_2 , N_3 et N_4) lui parviennent. Cependant, ces 3 neurones doivent émettre chacun un PA dans un ordre particulier, imposé par les valeurs du délai synaptique assignées à chacune des synapses (voir Table 3.3).

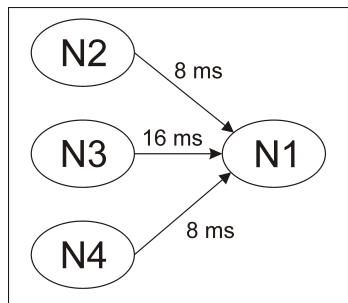


FIGURE 3.3 – Configuration d'un réseau de 4 neurones.

Paramètres du réseau "reconnaissance"								
$I_{stat}(N_1)$	$I_{stat}(N_2)$	$I_{stat}(N_3)$	$I_{stat}(N_4)$	w	p et t	τ_{AMPA}	Délai $_{N_2,4 \rightarrow N_1}$	Délai $_{N_3 \rightarrow N_1}$
-20	7	15	20	9	1	10 ms	8 ms	16 ms

TABLE 3.3 – Description du réseau de 4 neurones décrivant le réseau de reconnaissance temporelle. Les neurones utilisés sont des neurones RS. En condition initiale, les neurones N_2 à N_4 ont $v=-65$ et $u=-13$. Le neurone N_1 a $v=20$ et $u=-13$.

Le neurone N_1 devient ainsi un détecteur dont la séquence d'activation est composée par les PAs provenant de N_3 , puis N_2 et N_4 (i.e l'ordre des synapses ayant le délai synaptique le plus élevé au plus faible conformément à la configuration présentée dans le Tableau 3.3).

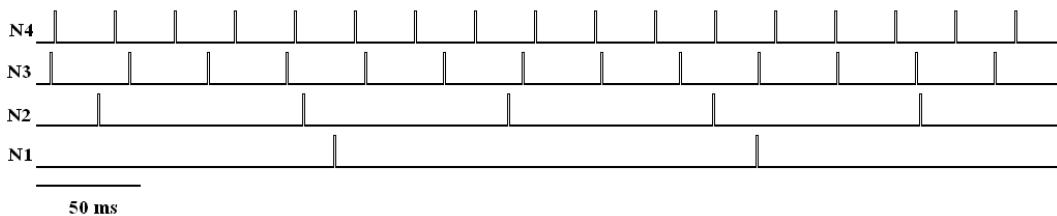


FIGURE 3.4 – Raster plot de l'activité des 4 neurones mettant en valeur la reconnaissance des motifs temporels.

La Figure 3.4 nous montre que le motif que l'on cherche à reconnaître est simple. Le neurone N_3 doit émettre un PA en premier, puis les neurones N_2 et N_4 doivent émettre simultanément¹ un PA. Si cet ordre est respecté, alors le neurone N_1 pourra émettre un PA.

¹Le réseau n'a pas été optimisé pour une telle application, ce qui explique la variabilité observée

3.2 Théorie des Central Pattern Generator

3.2.1 Approche connexionniste

Dans le règne animal, le mouvement (la marche chez la salamandre (Ijspeert et al., 2007), la nage chez les têtards (Li et al., 2014), le vol chez les criquets (Stevenson and Kutsch, 1987)) et même les battements de cœur (Hill et al., 2001)(Cymbalyuk et al., 2002) sont contrôlés par des CPGs. (Ijspeert, 2008) a démontré que les implémentations de CPGs sont divisées selon différents niveaux d'abstraction. Ces niveaux commencent par l'approche biophysique du modèle Hodgkin-Huxley (HH) (Hodgkin and Huxley, 1952), décrit en détail dans le Chapitre 1, et aboutissent par une approche plus abstraite (l'approche mathématique), basée sur les oscillateurs couplés (Zielinska, 1996). Entre ces 2 extrêmes se trouve l'approche "connexionniste", utilisant des modèles simples de neurones impulsionnels pour étudier les mécanismes spécifiques aux CPGs (Ekeberg, 1993).

Le but de l'approche biophysique est d'atteindre une proximité (ou même une équivalence) entre le système biologique et le modèle simulé afin de faciliter ou rendre possible l'expérimentation hybride. C'est pourquoi (Hill et al., 2001) rapporte les mesures effectuées sur un CPG biologique responsable du battement du cœur chez la sangsue (voir Table 3.4) avant d'utiliser le modèle HH pour émuler l'activité caractéristique de ce CPG. Comme nous l'avions vu au Chapitre 1, le modèle HH est peu souhaité pour une implémentation numérique.

L'approche abstraite se focalise principalement sur la structure des CPGs et les dynamiques de leurs phases. La neurorobotique utilise les CPGs pour coordonner les mouvements de robots bipèdes (Li et al., 2013), multipèdes (Barron-Zambrano and Torres-Huitzil, 2013) ou robots nageurs (Ijspeert et al., 2007). Les applications de la neurorobotique sont implémentées à l'aide de microcontrôleurs ou microprocesseurs (Crespi and Ijspeert, 2006), utilisant principalement des oscillateurs sinusoïdaux non-linéaires (Fitzhugh-Nagumo (FitzHugh, 1955)(FitzHugh, 1961)(Nagumo et al., 1962), Stein (Stein, 1967), Van der Pol (Der Pol, 1920)(Van der Pol and Van der Mark, 1928)). Les dynamiques décrites par ces systèmes ont été étudiées utilisant des outils mathématiques classiques (Zielinska, 1996) de manière à faciliter l'émulation des diverses mouvements trouvés dans le règne animal (comme le trot, la marche et le galop (Barron-Zambrano and Torres-Huitzil, 2013)). En dépit d'une possible implémentation du modèle Van Der Pol sur FPGA, démontré par (Barron-Zambrano and Torres-Huitzil, 2013), le niveau d'abstraction est trop élevé pour envisager une connexion avec des organismes vivants.

Une approche intermédiaire (dite "connexionniste") est nécessaire pour réaliser un CPG sur une plateforme FPGA via un modèle de neurone impulsional, faibles ressources, utilisable pour l'expérimentation hybride. Comme exposé dans le Chapitre 1, le modèle Izhikevich (IZH) (Izhikevich, 2003) possède les qualités requises pour cette implémentation. Afin de pouvoir implémenter le premier CPG utilisant le modèle IZH, nous avons d'abord étudié les CPGs biologiques dans la littérature et passé en revue les différentes implémentations de CPGs. Sur la base de l'architecture présentée dans le Chapitre 2, nous allons présenter notre CPG, sa topologie et les paramètres utilisés, en rappelant que notre objectif est de s'approcher le plus possible des organismes vivants. C'est d'ailleurs pour cela que, dans la suite de ce chapitre, nous allons comparer l'activité de notre implémentation de CPG avec l'activité mesurée dans le système biologique.

3.2.2 Principe de l'implémentation matérielle

La neurorobotique est un domaine scientifique combinant l'étude des neurosciences, la robotique et aussi l'intelligence artificielle. Ce domaine vise à étudier certains mécanismes biologiques (apprentissage, CPG et autres), à les formaliser et les implémenter. (Torres-Huitzil and Girau, 2008)(Rice et al., 2009)(Serrano-gotarredona et al., 2009)(Barron-Zambrano et al., 2010)(Barron-Zambrano and Torres-Huitzil, 2013) présentent des implantations de CPGs sur FPGA pour des applications robotiques. Ces CPGs sont fondés sur des modèles mathématiques de neurones simples. La priorité dans ces réalisations consiste à trouver l'inspiration dans les principes biologiques clés pour répondre aux problématiques d'ingénierie (systèmes bioinspirés) plutôt que d'imiter ou de s'approcher du comportement des organismes biologiques avec un esprit porté sur la compréhension des mécanismes régissant ce phénomène en vue d'expérimentations hybrides (systèmes biomimétiques).

D'ailleurs en robotique, les réseaux de neurones sans potentiel d'action (ces réseaux sont dépendants du temps : il ne s'agit pas de perceptrons) sont utilisés. Ces derniers sont basés sur le modèle Amari-Hopfield (Amari, 1972) ou d'autres modèles plus simples (Van der Pol and Van der Mark, 1928). En effet les CPGs utilisés pour la coordination des mouvements des robots multipèdes sont basés sur des oscillateurs sinusoïdaux avec des fréquences et amplitudes variables, à l'aide de plusieurs paramètres. Nous trouvons aussi plusieurs implantations analogiques réalisées par plusieurs équipes (Linares-Barranco et al., 1993)(Still and Tilden, 1998)(Lewis et al., 2001)(Nakada et al., 2003)(Lee et al., 2007)(Wijekoon and Dudek, 2008). Etant donné que nous cherchons à utiliser des réseaux de neurones impulsionnels, nous ne pouvons adopter aucune de ces solutions.

Au final, dans la robotique, émergent 2 politiques différentes sur l'implémentation et l'émulation de CPG qui sont :

- l'utilisation d'oscillateurs bioinspirés.
- l'implémentation de neurones bioinspirés sans complexité avancée.

Nous savons que des études (Jung et al., 2001)(Le Masson et al., 2002)(Vogelstein et al., 2006) ont réalisé une expérimentation hybride (i.e la connexion entre un réseau de neurones artificiels et un réseau de neurones biologiques). Les neurones artificiels mis en œuvre utilisaient le modèle HH (Hodgkin and Huxley, 1952).

(Hill et al., 2001) nous montre que le CPG Hodgkin-Huxley (CPG composé de neurones décrits par le modèle Hodgkin-Huxley (CPG-HH)) a été décrit à l'aide de plusieurs courants ioniques qui sont absents du modèle Izhikevich. La question est alors : "Comment décrire un CPG-IZH à partir de la description d'un CPG-HH ?"

3.2.3 Passage du CPG Hodgkin-Huxley au CPG Izhikevich

Pour effectuer le passage du CPG-HH au CPG-IZH, nous sommes passés par 4 grandes étapes. Premièrement, (Hill et al., 2001) nous présente comment utiliser des neurones HH et des synapses adaptées (modèle Hill) pour obtenir une activité électrique similaire aux CPGs biologiques, utilisant 10 conductances.

Indépendants, les neurones composant le CPG ont un comportement identique au neurone RS_{IZH} et ce n'est qu'une fois connectés par des synapses Hill que les neurones RS_{HH} entament un cycle de bouffées de PAs.

La complexité de la synapse joue un rôle majeur dans la génération et la réalisation des CPGs et celle-ci doit être adaptée au modèle de neurones qu'elle cherche à connecter. Évidemment, nous n'obtenons aucune activité en connectant des neurones IZH avec des synapses Hill.

(Nadim et al., 1995)(Olsen et al., 1995) présentent un modèle biophysique de cet oscillateur élémentaire en incluant des courants ioniques "mesurés" en voltage-clamp. En croisant différentes études (Nadim et al., 1995)(Olsen and Calabrese, 1996), (Hill et al., 2001) présente un nouveau modèle d'oscillateur élémentaire faisant intervenir des modifications intrinsèques des courants synaptiques.

Mais comment décrire un comportement issu des dynamiques des courants ioniques dans un modèle qui en est dépourvu ?

(Matsuoka, 1987) nous donne des éléments de réponse. En effet, dans cette étude, Matsuoka montre qu'il existe 4 stratégies pour générer un pattern rythmé :

- par une variation uniforme des stimuli d'entrée (voir Fig 3.7).
- par une variation temporaire du pattern des stimuli d'entrée du réseau permettant de produire potentiellement plusieurs rythmes.
- par l'altération des stimuli d'entrée.
- par la variation des poids synaptiques.

Pour répondre à cette question, il faut donc élever le niveau d'abstraction. Un neurone est un oscillateur qui n'est actif que lorsque le courant de stimulation lui parvenant est assez élevé. Ainsi un neurone appartenant à l'oscillateur élémentaire alternera entre un temps d'activité et un temps d'inactivité. Le temps d'inactivité du neurone N_2 est expliqué par l'inhibition exercée par le neurone N_1 qui, quant à lui, est dans son temps d'activité.

L'effet GABA-ergique permet de faire persister l'inhibition reçue sur le courant du neurone post-synaptique N_2 (ou N_1 si N_2 est le neurone pré-synaptique actif en premier) obligeant le neurone post-synaptique à être inhibé (donc inactif), puis lorsque l'inhibition s'estompera le neurone post-synaptique sera de nouveau actif. Cependant l'effet GABA-ergique n'explique pas à lui seul l'activité au sein d'un CPG. En effet, comme le montre (Matsuoka, 1987), il est nécessaire qu'une variation des poids synaptiques ait lieu pour démarrer et soutenir l'activité du CPG. C'est pour appliquer la quatrième stratégie que la plasticité à court-terme intervient. En suivant les explications données dans le Chapitre 2, plus un neurone pré-synaptique est actif, plus le(s) poids de la (des) synapse(s) conduisant au(x) neurone(s) post-synaptique(s) varie(nt) dans le temps suivant un paramètre P et une constante de temps τ_{syn} .

Au sein de notre CPG, en utilisant un paramètre P inférieur à 1 et une certaine constante de temps, plus le neurone N_1 sera actif, plus le poids de la synapse allant de N_1 vers N_2 sera diminué. Cela signifie que plus N_1 est actif, plus son emprise inhibitrice sur N_2 diminuera dans le temps, jusqu'à être très petite (voire négligeable). Dans le même temps, l'effet GABA-ergique va disperser l'inhibition subie par N_2 et permettre à N_2 de redevenir actif. Le secret de notre CPG numérique réside alors dans deux biomécanismes : l'effet GABA-ergique (défini par la constante de temps τ_{GABA}) et la plasticité à court-terme décrite par (Izhikevich and Edelman, 2008), définie par les paramètres x, p et τ_{syn} .

C'est ainsi que dans la quatrième et dernière étape de notre passage d'un CPG-HH à un CPG-IZH, nous utilisons des synapses munies de la plasticité à court terme, rendant un réseau de neurones IZH capable d'imiter le comportement des CPGs biologiques et des CPG-HH et de devenir par la même un CPG-IZH.

3.2.4 Topologie des CPGs

Bien qu'utilisant un modèle différent d'Izhikevich, (Matsuoka, 1987) présente également plusieurs CPG ayant différentes topologies. Par topologie, nous parlons de l'organisation d'un groupe de neurones c'est-à-dire, le nombre de neurones et le type de synapses qui les relient les uns aux autres. L'étude affirme alors que les CPGs nécessitent peu de neurones et décrit par conséquent des CPGs comprenant entre 2 et 6 neurones pour 2 à 18 synapses.

De plus, (Matsuoka, 1987) et (Hill et al., 2001) décrivent des CPGs comme des réseaux ayant une inhibition mutuelle. (Hill et al., 2001) va plus loin dans son étude en présentant le CPG Hodgkin-Huxley produisant une activité semblable au CPG biologique à l'origine du battement de coeur chez la sangsue. Cette étude nous montre alors que le CPG élémentaire est composé de 2 neurones (proche de la description RS donnée par Izhikevich) reliés mutuellement par 2 synapses inhibitrices.

Ces études conjointes vont donc conditionner notre expérimentation, à savoir que notre CPG élémentaire sera composé de 2 neurones RS connectés par 2 synapses inhibitrices ayant la plasticité à court-terme. (Hill et al., 2001) présente également un CPG segmental composé de 8 neurones et des synapses inhibitrices. Nous allons donc dans un premier temps étudier le CPG élémentaire, puis dans un deuxième temps le CPG segmental.

Topologie et activité du CPG élémentaire

La Figure 3.5 présente la topologie et l'activité caractéristique de l'oscillateur élémentaire. Ce dernier a été obtenu en utilisant des neurones RS ($a=0.02$, $b=0.2$, $c=-65$ et $d=8$) et les paramètres suivants. Le poids synaptique des 2 synapses est égal à -4, le pourcentage P est égal à 10%. Les constantes de temps τ_{GABA} et τ_{syn} sont respectivement égales à 2000 ms et 10 000 ms.

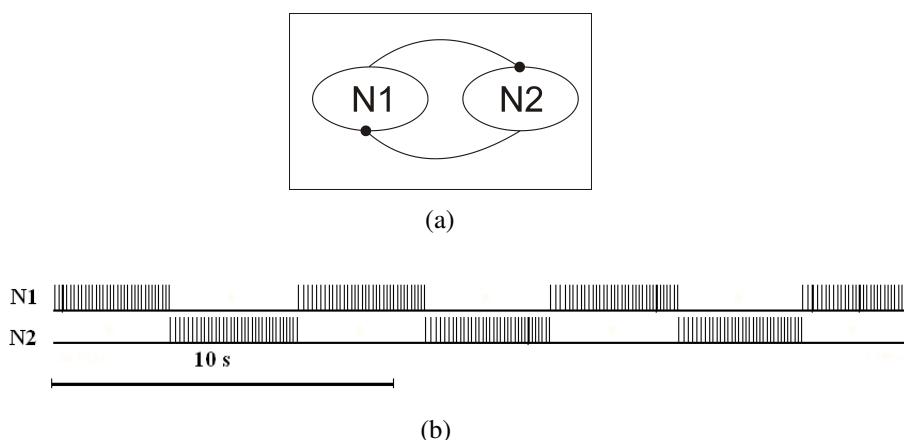


FIGURE 3.5 – Présentation de la topologie de l'oscillateur élémentaire composé de 2 neurones RS et 2 synapses inhibitrices (Sous-Figure (a)) et son activité caractéristique (Sous-Figure (b)).

Topologie et activité du CPG segmental

L'oscillateur segmental est composé de 8 neurones RS relié par 12 synapses inhibitrices identiques, dont 2 paires motrices (N_1/N_2 et N_3/N_4), et les neurones N_5 à N_8 sont des neurones permettant de stabiliser l'activité des paires motrices (Figure 3.6). La présence de neurones

supplémentaires permet une meilleure stabilité du CPG et permettent aux bouffées de PA d'apparaître sur une plus large plage de périodes.

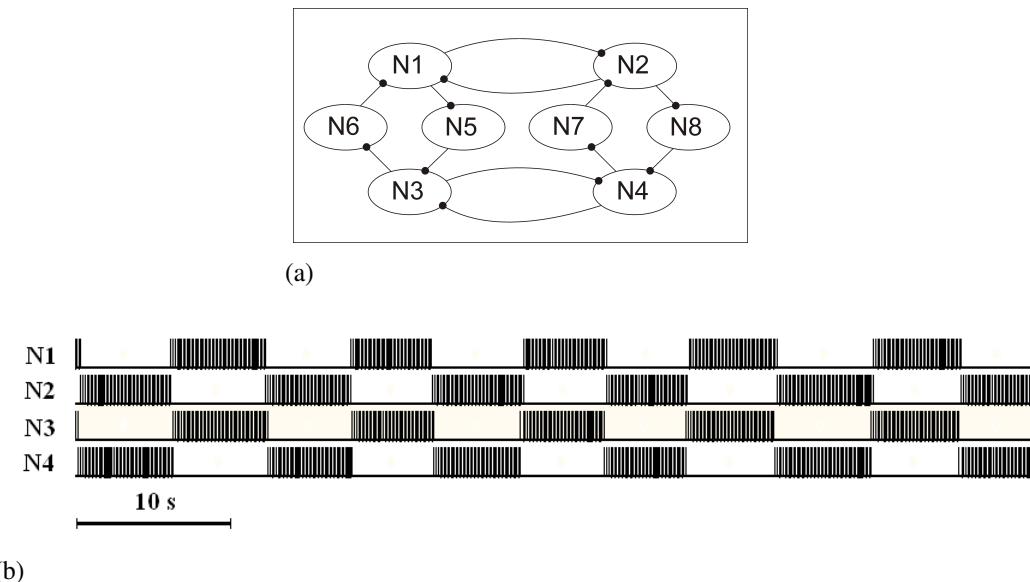


FIGURE 3.6 – Présentation de la topologie de l'oscillateur segmental (Sous-Figure 3.6(a)) et l'activité caractéristique des paires motrices N_1/N_2 et N_3/N_4 (Sous-Figure 3.6(b)).

Comparaison entre notre implémentation et le système biologique

La Table 3.4 présente une comparaison entre le système biologique et les CPG numériques. Cette comparaison est similaire à celle trouvée dans (Hill et al., 2001). Nous avons pris le système biologique pour référence et sélectionné un CPG numérique de chaque type (oscillateur élémentaire et oscillateur segmental) ayant une activité périodique proche. Ces résultats ont été obtenus en utilisant des neurones RS ($a=0.02$, $b=0.2$, $c=-65$ et $d=8$) et les paramètres suivants :

- pour l'oscillateur segmental. Le poids des synapses reliant les neurones N_1/N_2 et N_3/N_4 est égal à -4 et leurs pourcentage est égal à 1%. Les autres synapses partant des neurones N_1 , N_2 , N_3 et N_4 ont un poids synaptique égal à -5.2 et un pourcentage de 0.1%. Les synapses des neurones de coordination N_5 , N_6 , N_7 et N_8 ont un poids de -1.5 pour un pourcentage de 0.1%. Pour finir, les constantes de temps τ_{GABA} et τ_{syn} sont respectivement égales à 1000 ms et 5000 ms.

Dans le cadre du projet HYRENE, trois acteurs interviennent : la plateforme numérique avec notre CPG numérique, le stimulateur piloté par le CPG et la moelle épinière sectionnée. Ce stimulateur comporte deux voies d'entrées numériques permettant de déclencher l'envoi des stimuli électriques pré-programmés. C'est pourquoi nous extrayons de notre CPG numérique deux PA qui seront utilisés comme déclencheur.

En procédant ainsi, les propriétés importantes de notre CPG sont la période moyenne et le rapport cyclique moyen. Par ailleurs, nous pouvons voir dans la Table 3.4 que les périodes moyennes et les rapports cycliques moyens des oscillateurs numériques sont très proches de la biologie.

Notre approche connexionniste nous permet d'obtenir une période moyenne et un rapport cyclique moyen proche de la biologie. Ceci nous permet de déclencher les stimuli électriques avec un timing compatible avec le vivant. La contrepartie de notre approche ne permet pas d'égaler les fréquences mesurées dans les bouffées dans le système biologique.

	Système Biologique	Oscillateur numérique Élémentaire	Oscillateur numérique Segmental
Période moyenne	11 ± 1 s	11.9 ± 0.1 s	11.5 ± 1 s
Rapport cyclique moyen	$57.2 \pm 2.9\%$	49.2-50%	$56.5 \pm 4\%$
Fréquence moyenne intra-burst	11.9 ± 2.1 Hz	8.7 ± 2.7 Hz	9 ± 2 Hz
Fréquence initiale moyenne	4.3 ± 0.7 Hz	6.1 ± 0.2 Hz	6.7 ± 0.2 Hz
Fréquence maximum moyenne	17.5 ± 3.2 Hz	10 Hz	11 Hz
Fréquence finale moyenne	5.8 ± 1.0 Hz	8.3 ± 0.4 Hz	9.0 ± 0.5 Hz

TABLE 3.4 – Comparaison entre les oscillateurs numériques et le système biologique.

Vers un CPG multimodal : Etude de la stimulation d'un CPG

Sans stimulation, les CPGs présentés, dans la Sous-section 3.2.4, sont des CPGs avec un seul rythme (ou mode). Est-il alors possible de générer plusieurs rythmes à partir d'un CPG qui n'en possède intrinsèquement qu'un seul ? Comment est-il possible d'obtenir un CPG multimodal ? Dans cette sous-section, nous nous proposons d'étudier le CPG multimodal.

Paramètres pour les oscillateurs élémentaires							
I _{stat}	w _{N1 → N2}	w _{PA_x → N_x}	p _{N1 → N2}	τ _{syn}	τ _{GABA}	τ _{AMPA}	Période (en s)
7	-5.2	9	0.1	1 000 ms	5 000 ms	100 ms	10

TABLE 3.5 – Paramètres de l'oscillateur élémentaire et sa connectivité externe.

Plus précisément, nous allons étudier l'influence de stimuli externes sur le comportement d'un CPG. Pour nous mettre en contexte dans le projet HYRENE, ces stimuli externes peuvent être tout simplement des signaux issus ou déduits d'un réseau de neurones biologiques, considérés comme des commandes de l'activité du CPG. Pour illustrer cette stratégie, nous allons utiliser un oscillateur élémentaire ayant les caractéristiques décrites dans la Table 3.5. Pour produire la Figure 3.7, l'oscillateur élémentaire sera stimulé par des PAs générés à l'aide d'un compteur.

La Figure 3.7 nous montre le CPG élémentaire présenté précédemment (avec une période de 10 s). Lors de cette expérience, nous injectons à l'entrée de ce CPG un vecteur de PA qui suit la condition suivante. À l'aide d'un compteur, les neurones N₁ et N₂ reçoivent chacun un PA (respectivement PA₁ et PA₂). Le PA externe envoyé à N₂ est émis entre 2 PAs externes envoyés à N₁. Les potentiels d'action PA₁ et PA₂ sont transmis respectivement aux neurones N₁ et N₂ par une synapse excitatrice (avec w_{PA_x → N_x}=9, p_{PA_x → N_x}=1, τ_{syn(PA_x → N_x)}=1 ms).

Cette expérience met en œuvre une stratégie évoquée par Matsuoka pour obtenir un CPG multimodal en utilisant l'influence de stimuli (ou PAs) appliqués aux neurones composant le CPG. Par extension, il est possible d'appliquer ce résultat sur les paires directrices du CPG segmental.

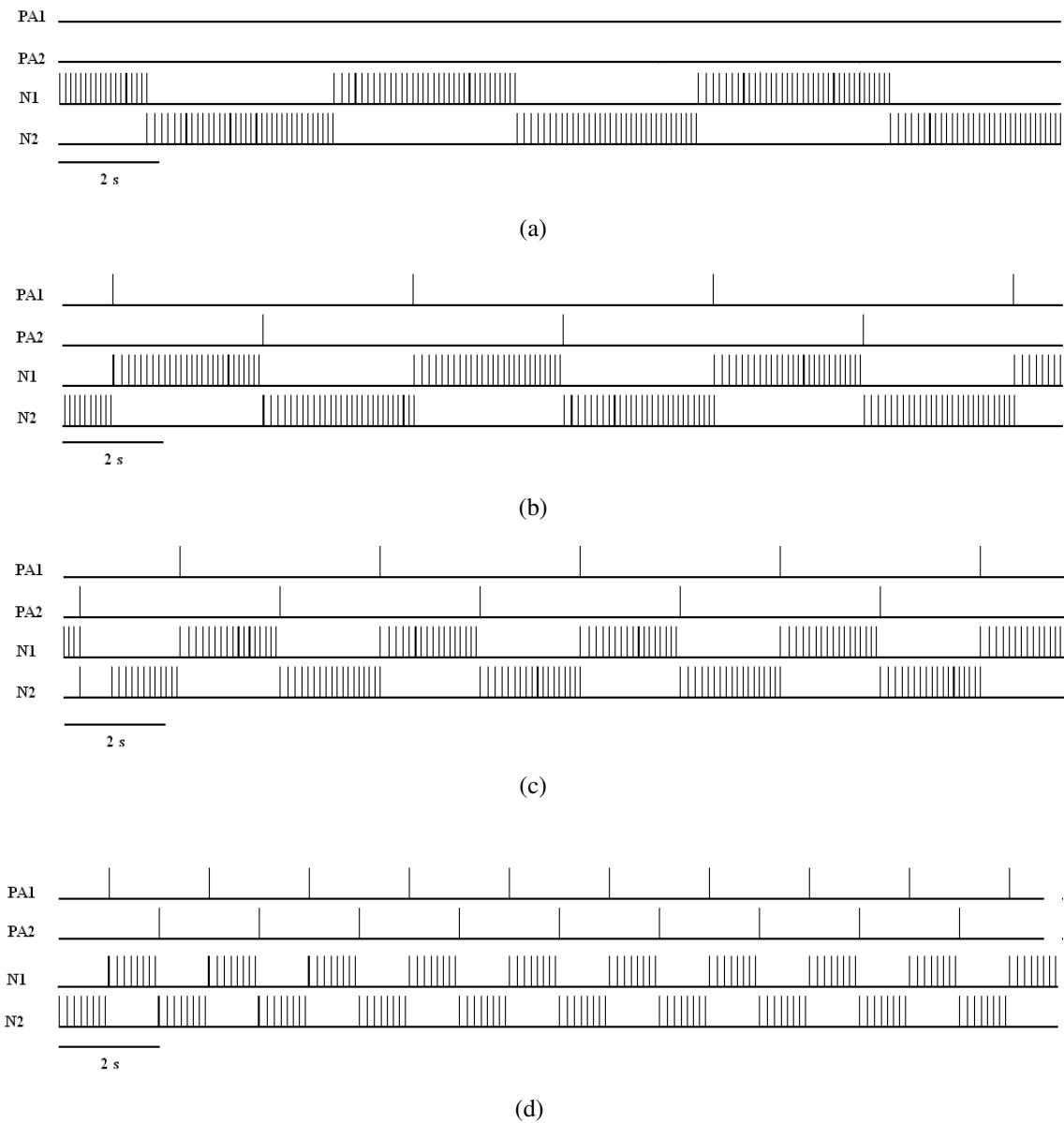


FIGURE 3.7 – Influence des stimuli externes (PA1 et PA2) sur un CPG élémentaire. La Sous-Figure (a) présente un oscillateur élémentaire sans stimuli externes. Les autres cas présentent une stimulation sur les neurones (N1 et N2) allant de 3 secondes d'écart (Sous-Figure (b)), 2 secondes d'écart (Sous-Figure (c)), 1 seconde d'écart (Sous-Figure (d)) entre chaque stimuli. Chaque stimulation est exercée sur le même CPG présenté en (a).

En stimulant les neurones d'un oscillateur élémentaire, il est possible de changer la dynamique du CPG comme présenté dans la Figure 3.7 et ainsi lui faire adopter des périodes alternatives à sa période d'origine.

3.2.5 Réglage de la période des bouffées de potentiels d'actions

En changeant les paramètres τ_{GABA} et τ_{syn} , il est possible d'obtenir d'autres CPG ayant des périodes différentes. Pour comprendre l'influence de ce couple de paramètres, nous avons

commencé par changer ces 2 paramètres au sein d'un oscillateur élémentaire. La Table 3.6 montre l'influence de ce couple de paramètres sur la période d'un oscillateur élémentaire. Dans le projet HYRENE, nous sommes intéressés par des CPGs ayant une période allant entre 1 à 10 s.

Malgré cette technique de division par 2 (Table 3.6), l'oscillateur élémentaire n'est pas suffisant pour obtenir un CPG avec une période proche de la seconde. Pour atteindre ces valeurs, il nous faut donc augmenter le nombre de neurones pour nous approcher de cette valeur. C'est ainsi qu'en s'inspirant de (Hill et al., 2001), nous avons étudié l'oscillateur segmental (CPG à 8 neurones).

Paramètres pour l'oscillateur élémentaire					
I_{stat}	w	p	τ_{syn}	τ_{GABA}	Période moyenne (en s)
7	-5	0.01	5 000 ms	1 000 ms	4.6
			10 000 ms	2 000 ms	13.6
			20 000 ms	4 000 ms	30.5

TABLE 3.6 – Evolution de la période au sein de l'oscillateur élémentaire en fonction des paramètres du réseau de neurones numérique (poids synaptiques, τ_{GABA} et τ_{syn}).

La Table 3.7 nous montre qu'en utilisant un oscillateur segmental, tout en faisant varier le couple τ_{syn} , τ_{gaba} , il est à présent possible d'obtenir un intervalle de période allant de 1.3 s à 10 s.

Paramètres pour l'oscillateur segmental						
I_{stat}	$W_{N1 \leftarrow N2}$ $W_{N3 \leftarrow N4}$ $W_{N1 \rightarrow N5}$ $W_{N2 \rightarrow N8}$	$W_{N5 \leftarrow N3}$ $W_{N6 \rightarrow N1}$ $W_{N7 \rightarrow N2}$ $W_{N8 \rightarrow N4}$	p	τ_{syn}	τ_{GABA}	Période moyenne (en s)
7	-4	-1.5	0.1	5 000 ms	1 000 ms	10
	-4.8	-1.5		2 500 ms	500 ms	3
	-4	-0.4		1 250 ms	250 ms	1.3

TABLE 3.7 – Evolution de la période au sein de l'oscillateur segmental en fonction des paramètres du SNN (poids synaptiques, τ_{GABA} et τ_{syn}).

3.3 Résultats expérimentaux des différents projets

3.3.1 Résultats du projet BRAINBOW

Présentation générale du projet

la Figure 3.8 illustre notre présentation du projet BRAINBOW afin d'en faciliter la compréhension. Nous utiliserons également des photos prises avec l'aimable accord du Dr Michela Chiappalone et présentées dans la Figure 3.9.

L'objectif du projet BRAINBOW est d'obtenir une preuve de concept d'une neuroprothèse neuromorphique pouvant réparer des lésions corticales par jonction ou substitution de la zone endommagée. Pour atteindre cet objectif, le projet BRAINBOW passe par l'étude des réseaux de

neurones en 2D et en 3D, mais aussi l’élaboration d’un réseau de neurones numérique capable d’imiter une activité d’un réseau de neurones biologiques.

La Figure 3.8 résume l’idée globale des expérimentations du projet BRAINBOW, qui est de mettre en œuvre notre réseau de neurones configurable implanté dans la plateforme MultiMed (permettant l’enregistrement et la détection des PAs biologiques), un stimulateur MCS 2 voies et des neurones posés sur des électrodes (MEA 64 voies MCS équipés de pré-amplificateurs MCS) (Figure 3.8). Notre réseau de neurones sera configuré en début d’expérience selon les paramètres trouvés par l’UNIGE (équipe de modélisation au sein du projet). Le stimulateur a été programmé pour émettre des stimulations électriques (série d’impulsions biphasiques, d’une amplitude de 750 mV pendant 300 μ s) à la réception de signaux déclencheurs (ici PAs provenant de notre réseau de neurones). Les neurones sont disposés en une couche uniforme sur les électrodes et séparés en quatre groupes (ou clusters) de neurones, voir la Figure 3.9(a). Les cultures de neurones utilisées pour nos expérimentations ont été réalisées par l’IIT (Istituto Italiano di Technologia). Pour finir, nous disposons d’un système Matlab pour décoder et traiter les données offline.

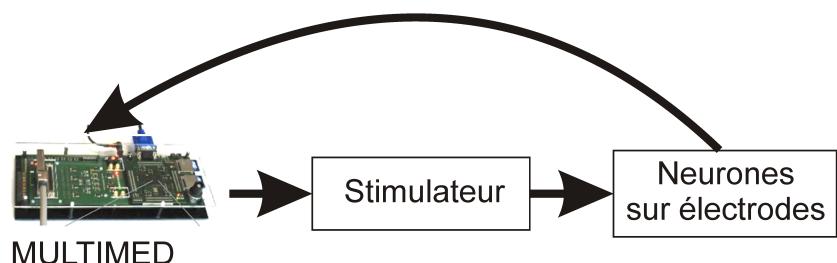
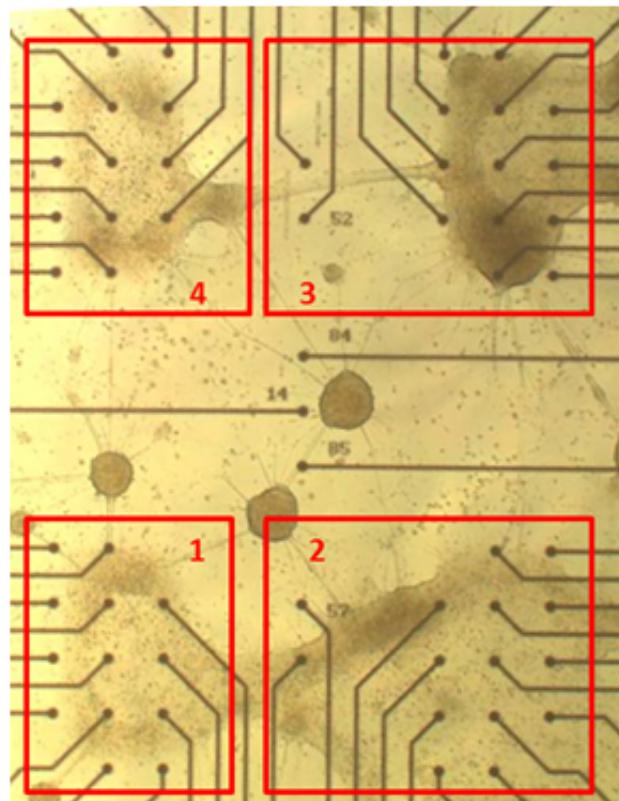


FIGURE 3.8 – Vue globale du projet BRAINBOW. Ce projet met en œuvre Multimed (plateforme numérique sur laquelle est implémenté notre réseau de neurone), un stimulateur commercial préprogrammé et des neurones posés sur un lit d’électrodes.

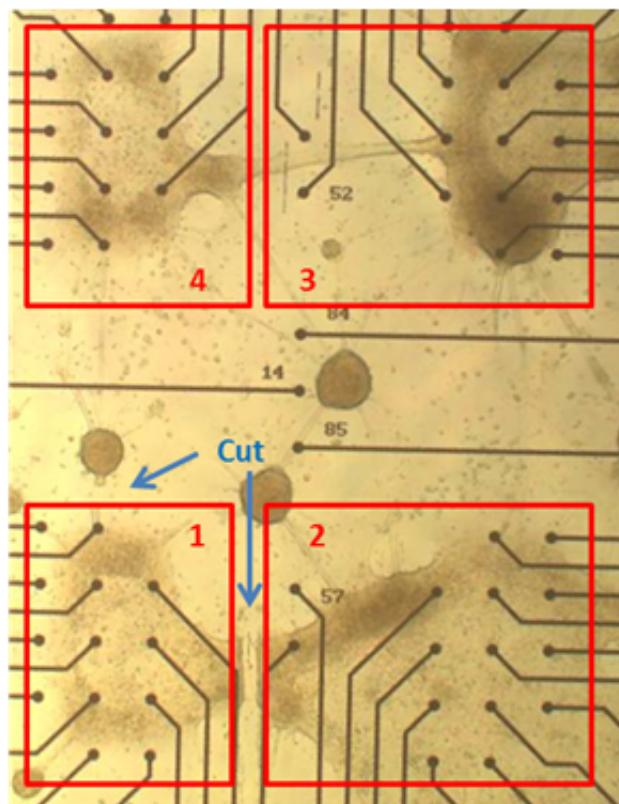
Les 4 Clusters et la lésion

Les expérimentations BRAINBOW démarrent avec des neurones cultivés sur des électrodes. Sur la Figure 3.9, nous pouvons distinguer 4 clusters de neurones, encadrés en rouge. Un cluster de neurones est un groupe de neurones interconnectés. Chaque cluster possède un groupe de synapses lui permettant de se connecter aux autres clusters. Le projet BRAINBOW a notamment montré une méthode efficace présentée par le TAU pour identifier les neurones par l’imagerie calcium aussi bien au sein d’un cluster de neurones que dans des portions de cerveau (Bonifazi et al., 2013).

La Figure 3.9 (partie droite) nous montre un exemple de lésion (ici faite au laser par l’IIT) autour du cluster 1. Cette lésion va sectionner les axones connectant ce cluster aux autres et ainsi isoler le cluster 1.



(a)



(b)

FIGURE 3.9 – Photos de clusters de neurones sur MEA avant et après une lésion. Les neurones sont regroupés en cluster (repéré par les cadres rouges). Chaque cluster est connecté à ses voisins par des synapses visibles sur la photo. (a) montre ces clusters avant la lésion et (b) montre les mêmes clusters après une lésion opérée au laser (repérée par les flèches bleues). La distance entre 2 électrodes est de 200 μm .

Le 5e cluster : le réseau de neurones artificiels

Nous allons ensuite introduire un cinquième cluster : notre réseau de neurones artificiel (RNA). Le cluster 5 a été configuré pour reproduire un comportement d'un cluster de 100 neurones biologiques. Les paramètres de ce réseau nous ont été donnés par l'UNIGE (Table 3.8), une équipe partenaire dans le projet BRAINBOW. Ce réseau a pour spécification les fréquences des épisodes de synchronicité similaires aux données biologiques enregistrées.

Notre réseau possède 100 neurones (80 neurones excitateurs et 20 neurones inhibiteurs), une connectivité moyenne de 50%, et dont chaque synapse possède une plasticité synaptique décrite dans la Table 3.8. La Figure 3.10 montre le type d'activité retenue pour notre réseau de neurones, dans laquelle nous pouvons observer des épisodes de synchronicité (moment où la majorité, ou même la totalité, des neurones émettent des PAs simultanément).

Cependant, dans cet exemple, notre réseau a été adapté aux fréquences présentes dans les clusters (autour de 0,14 Hz (Bonifazi et al., 2013)) et possède une activité plus lente que celle présente dans le cortex. Dans nos expérimentations, ce cluster permet de rétablir la communication entre le cluster isolé (comme présenté dans la Section précédente) et les autres clusters, par exemple.

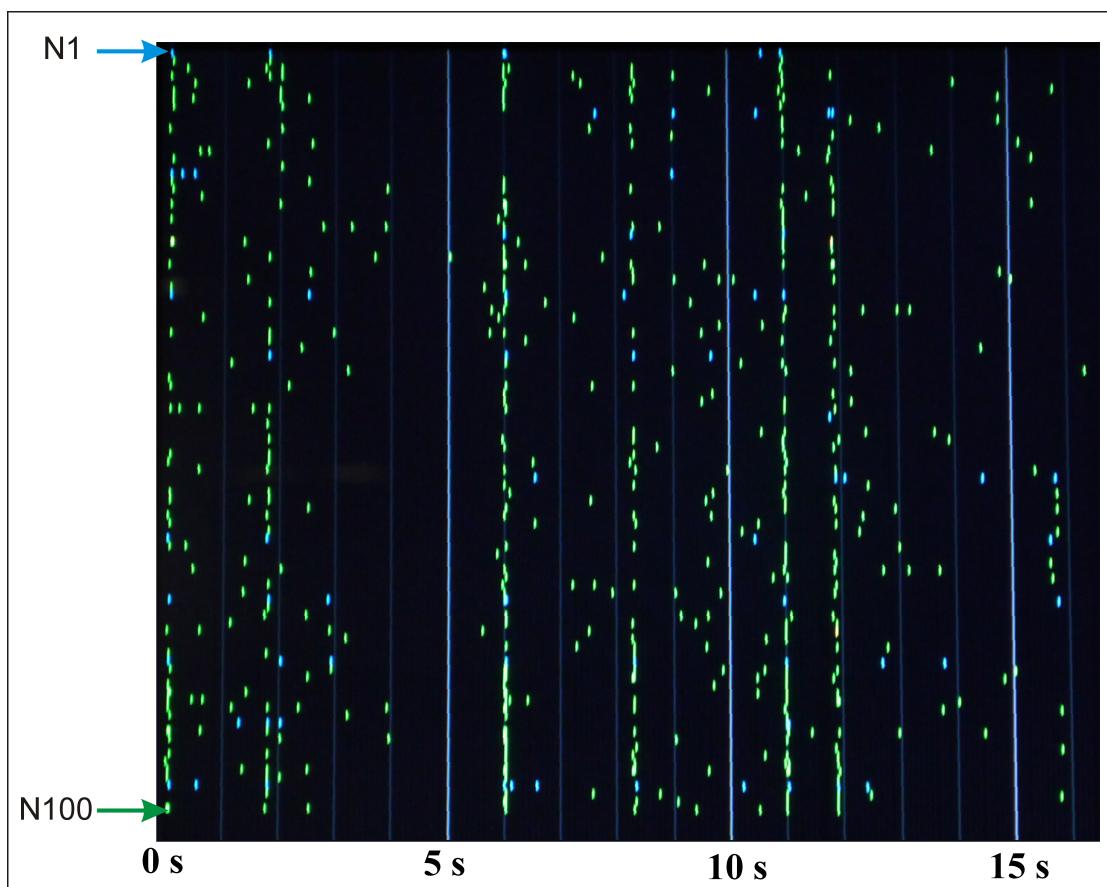


FIGURE 3.10 – Exemple d'activité du réseau de 100 neurones soumis au bruit généré par le processus d'Ornstein-Uhlenbeck. La connectivité des neurones est de 50 synapses plastiques par neurone en moyenne. Nous pouvons voir les PAs émis par les 100 neurones (points verts et bleus). L'activité est caractérisée par des épisodes de synchronicité (émission simultanée de PAs par la majorité des neurones).

A) Condition initiale des 80 neurones excitateurs							
a	b	c	d	I_{stat} , I_{exc} et I_{inh}	u	v	μ
0.02	0.2	entre -50 et -65	entre 5 et 8	0	-13	-65	0
B) Condition initiale des 20 neurones inhibiteurs							
a	b	c	d	I_{stat} , I_{exc} et I_{inh}	u	v	μ
0.1	0.2	-65	2	0	-13	-65	0
C) Résumé de la connectivité interne du réseau et sa condition initiale							
Connectivité moyenne		W_{syn}		P	t_{syn}	X_{syn}	Délai (ms)
exc.	50 synapses par neurone	entre 0.2 et 3.5		0.6	0.0067	1	1 ou 2
inh.		entre -0.002 et -4					
D) Résumé de la connectivité externe du réseau et sa condition initiale							
Connectivité		W_{syn}		P	t_{syn}	X_{syn}	Délai (ms)
1 synapse par neurone		9		0.6	0.0067	1	1

TABLE 3.8 – Résumé des paramètres permettant de créer l’activité du réseau dans la Figure 3.10. Le fichier de configuration pour ce réseau étant conséquent, nous avons choisi de présenter le principe de base de ce réseau. Les 100 neurones sont divisés en 2 groupes : 80 neurones excitateurs (Table A) et 20 neurones inhibiteurs (Table B) dont la connectivité des 2 groupes est repérée par respectivement la ligne "exc." et la ligne "inh." de la Table C. La Table D indique la connectivité externe du réseau. Celle-ci est en liaison directe (le premier PA provenant de l’extérieur est connecté au neurone N₁, le deuxième PA provenant de l’extérieur à N₂, ainsi de suite pour le centième PA connecté à N₁₀₀).

Les premiers résultats² du projet BRAINBOW

Après avoir présenté les clusters sur les MEAs et la lésion, nous allons présenter un scénario d'expérimentation hybride (Figure 3.11). Cette figure simplifiée comporte la succession d'expériences effectuées sur les mêmes clusters de la Sous-Figure 3.11(a) à la Sous-Figure 3.11(d).

L'échelle de couleurs³ représente l'entropie du transfert de données allant du bleu très foncé (décrivant un échange de données très faible, voire inexistant) au rouge foncé (décrivant un échange très important), quantifié en bit/s. L'entropie du transfert de données va permettre de modéliser l'impact de l'activité d'un cluster sur l'activité d'un cluster auquel il est relié (par exemple l'impact de l'activité du cluster 2 sur l'activité du cluster 3).

Le sens de la connexion est schématisé par une flèche, aboutissant sur un carré coloré. Suivant l'échelle de couleur présenté, plus la couleur est proche du rouge et plus le cluster en amont de la flèche a un impact sur le cluster en aval.

Nous avons choisi de présenter un scénario de jonction simple, où le RNA va remplacer le cluster 1 (Figure 3.9(b)) et transmettre une activité du cluster 2 au cluster 4.

La Sous-figure 3.11(a) présente une situation initiale située avant la lésion. La mesure de l'entropie nous montre que le cluster 5 (RNA) est isolé et n'a aucun impact sur les clusters 2 et 3 qui, eux à leur tour, n'ont aucun impact sur le cluster 5. Cette mesure est matérialisée par la couleur bleu foncé des carrés. Nous pouvons remarquer que le cluster 2 et le cluster 3 entraînent chacun une activité chez le cluster auquel ils sont connectés (bleu et bleu clair). En résumé, la Sous-figure 3.11(a) nous présente une situation initiale dans laquelle seuls les clusters 2 et 3 sont interconnectés (ou en liaison directe).

Enfin, la Sous-Figure 3.11(b) présente une situation où une lésion a séparé physiquement le cluster 2 et le cluster 3. Les clusters 2 et 3 n'ont plus aucun impact l'un sur l'autre (repérée par le bleu foncé). La lésion empêche ces deux clusters de communiquer.

Dans ce scénario (Sous-Figure 3.11(c)), nous allons recréer une communication entre le cluster 2 qui sera le cluster "trigger" et le cluster 3 qui sera notre cluster "cible", par le biais du cluster 5 (notre RNA configuré), créant ainsi une liaison indirecte entre le cluster 2 et le cluster 3. Par un effet de chaîne, l'activité du cluster 2 permet d'entrainer l'activité du cluster 5 (passage du bleu foncé à rouge foncé) et l'activité du cluster 5 va entraîner l'activité du cluster 3 (passage du bleu foncé au bleu clair). En mesurant l'entropie de transfert allant du cluster 2 au cluster 3, nous remarquons que celle-ci a augmenté (passage du bleu foncé à bleu clair), montrant ainsi que l'activité du cluster 2 (cluster "trigger") entraîne une activité dans le cluster 3 (cluster "cible") et donc que la liaison indirecte a été réalisée avec succès.

En résumé, la Sous-Figure 3.11(c) montre qu'il est possible de rétablir une communication entre le cluster 2 et le cluster 3 au travers de notre RNA, suite à une lésion.

La Sous-Figure 3.11(d) est une expérience témoin dans laquelle nous retirons le cluster 5, reproduisant la situation vue dans la Sous-Figure 3.11(b).

Pour conclure, (Guggenmos et al., 2013) a montré dans son étude la possibilité d'implanter une neuroprothèse non neuromorphe capable d'être l'intermédiaire entre deux zones cérébrales. La Figure 3.11 nous montre qu'il est possible d'utiliser un réseau de neurones artificiel pour faire la jonction entre deux réseaux de neurones biologiques, montrant par la même occasion qu'une meilleure compréhension des réseaux de neurones nous a permis d'interfacer l'artificiel

²Ces résultats ont été reproduits plusieurs fois. Cela reste des expériences préliminaires qui seront complétées ultérieurement.

³Pour des raisons de compréhension, un code de couleur a été choisi pour représenter l'entropie du transfert.

et le vivant. En effet, en connectant le cluster 2 au cluster 5 et connectant celui-ci au cluster 3, il est possible de rétablir la communication entre le cluster 2 et le cluster 3.

Des expérimentations en novembre 2015 viendront approfondir ces premiers résultats pour montrer que l'utilisation du RNA peut aussi servir à moduler l'activité au sein d'un réseau de neurones biologiques.

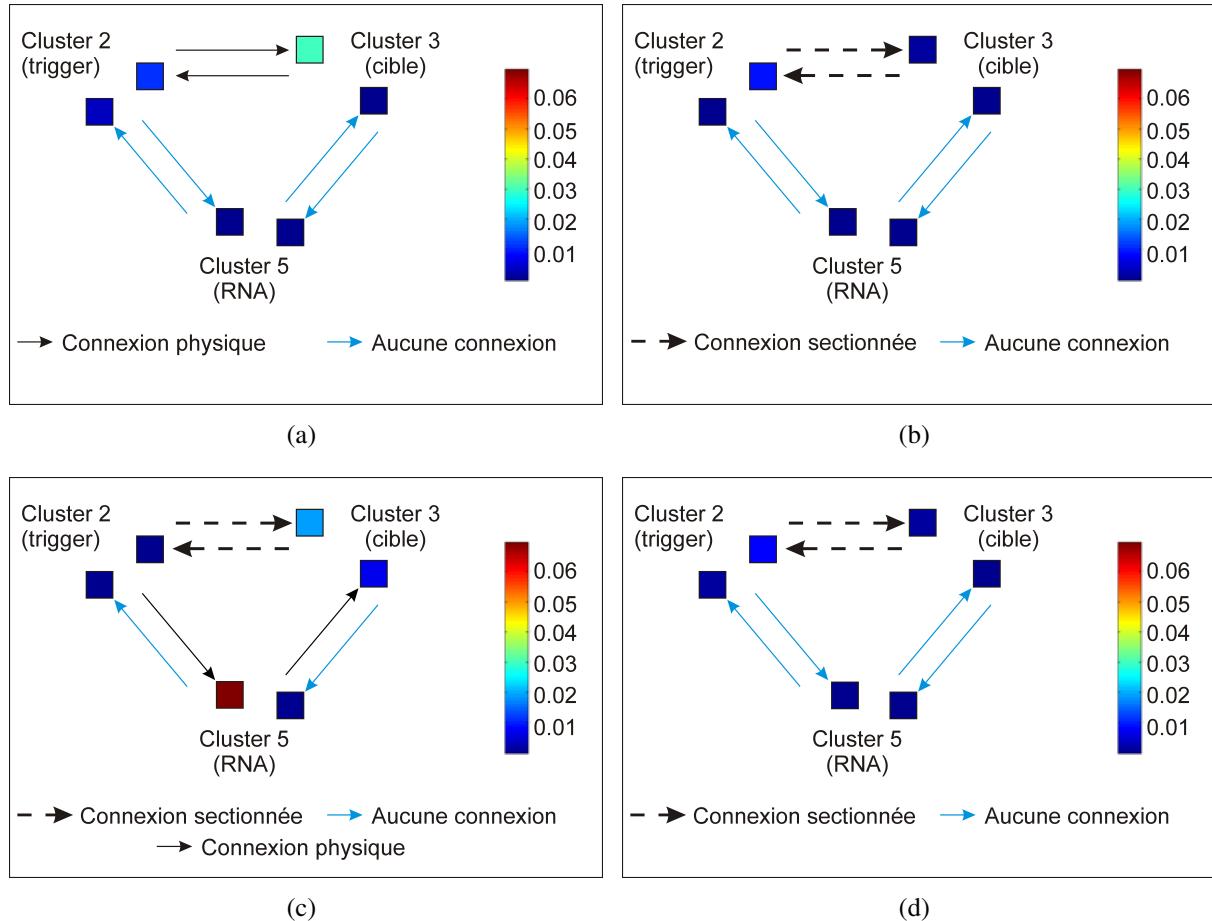


FIGURE 3.11 – Procédure expérimentale. Chaque flèche représente une connexion et se termine sur un carré qui représente l'activité du réseau induite par la connexion. Plus l'activité du cluster est élevée et plus la couleur s'approche du rouge. La Sous-Figure (a) représente une situation initiale où les clusters 2 et 3 sont connectés. La Sous-Figure (b) représente la lésion visant à séparer le cluster 2 et le cluster 3. Pour rétablir l'activité du cluster 3, le cluster 5 (RNA) va servir de jonction entre le cluster 2 et le cluster 3 (Sous-Figure (c)). Dans la Sous-Figure (d), le cluster 5 est déconnecté et aucune activité n'est transmise.

3.3.2 Résultats du projet HYRENE

Présentation générale du projet

Pour faciliter la compréhension de l'ANR Hyrène, nous avons introduit des photos de la moelle épinière de rat, réalisées avec l'aimable accord du Dr Blaise Yvert (CEA) dans la Figure 3.12. Dans une moelle épinière (ici celle d'un rat), les racines ventrales antagonistes L2/L5 génèrent les bouffées de PAs nécessaires pour entraîner la locomotion du rat. Etant donnée que

le rat est un quadrupède, nous repérons 4 sites de locomotion (L2R/L2L, pour le contrôle des pattes avant respectivement droite et gauche et L5R/L5L (Figure 3.12(a)), pour le contrôle des pattes arrière droite et gauche).

Les bouffées alternées de notre CPG correspondent à une séquence flexeur-extenseur. En régime normal, la moelle épinière intacte transmet les instructions provenant du cerveau aux racines ventrales L2/L5, qui entraînent à leur tour la locomotion. Si la moelle épinière est sectionnée transversalement (Figure 3.12(b)), les instructions provenant du cerveau sont stoppées et ne peuvent plus être transmises aux racines ventrales. Les racines L2/L5 deviennent inactives.

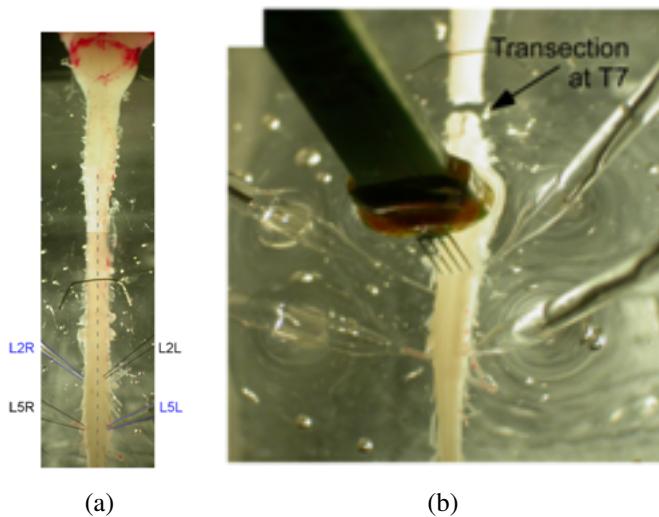


FIGURE 3.12 – Photos de moelle épinière de rat utilisé lors des expérimentations hybrides. La Sous-Figure (a) montre la moelle épinière, repérant les racines L2/L5 et la Sous-Figure (b) montre partant du haut une section transversale (qui est la lésion opérée sur la moelle épinière), l'électrode en fourchette et nos racines L2/L5 à sa base.

La Figure 3.13 résume l'idée de ce projet qui est d'étudier la faisabilité d'une microstimulation électrique pilotée par notre CPG numérique (fourni par l'équipe AS2N et embarqué sur Multimed appartenant à l'équipe Elibio). Cette microstimulation (composée de 10 impulsions biphasiques, chacune séparée d'1 ms, fournissant 150-300 μ A pendant 500 μ s) sera délivrée, par un stimulateur MCS 2 voies, sur les sites appropriés au travers d'une électrode en fourchette. Cette électrode en fourchette a été créée spécialement pour ce projet par l'ESIEE (Paris) connectée au système BioMea (Charvet et al., 2007), afin de permettre l'enregistrement de l'activité électrique issue de la colonne vertébrale et aussi la stimulation très localisée de celle-ci (Joucla et al., 2013). Les préparations de la moelle épinière (localisation des racines et des sites à stimuler, la section transversale et le maintien des tissus biologiques) incombaient à l'équipe du Dr Yvert (INCIA, Bordeaux).

Premiers résultats du projet Hyrène

Après avoir expliqué la nature de notre CPG et son activité (Section 3.2.4), nous allons passer au CPG final implémenté dans MULTIMED. Le stimulateur utilisé dans ce projet possède deux voies d'entrée numériques permettant de déclencher les microstimulations voulues. Par conséquent, nous avons choisi d'utiliser l'activité d'une paire motrice de notre CPG pour déclencher les microstimulations.

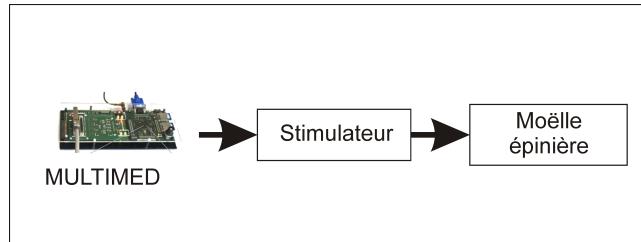


FIGURE 3.13 – Vue globale du projet HYRENE. Ce projet met en œuvre Multimed (plateforme numérique sur laquelle est implémenté notre CPG), un stimulateur préprogrammé et une moelle épinière.

En effet, nous avons vu que le CPG a une activité en bouffées en antiphase, reprenant l’activité en alternance entre les racines du côté droit et du côté gauche de la colonne vertébrale. Cependant, il n’est pas nécessaire d’utiliser chaque PA de notre CPG numérique comme déclencheur d’une microstimulation. Une succession rapide de microstimulations (de l’ordre de la milliseconde) risquerait d’endommager prématurément les tissus biologiques et l’électrode. Nous avons par conséquent décidé d’extraire le premier PA de chaque bouffée provenant d’une paire motrice (par exemple les neurones N3/N4 dans la Figure 3.14). Celui-ci déclenchera alors la microstimulation adéquate à la racine ciblée (Joucla et al., 2013).

La dernière étape pour construire le CPG final du projet HYRENE est de déterminer la méthode d’extraction du premier PA des bouffées qui nous intéressent. Pour éviter d’utiliser une méthode basée sur des compteurs numériques (qui serait à recalibrer lors de changements de fréquence du CPG), nous allons utiliser notre réseau de neurones pour créer les signaux déclencheurs désirés. En utilisant une propriété de notre plasticité synaptique (présentée en Section 3.1.2), il est possible alors de stimuler fortement un neurone (non présent dans le CPG présenté en Section 3.2.4) grâce au premier PA d’une bouffée. La Table 3.9 nous donne les paramètres des synapses aboutissant sur les neurones NR/NL et les paramètres de ces neurones (voir Figure 3.14).

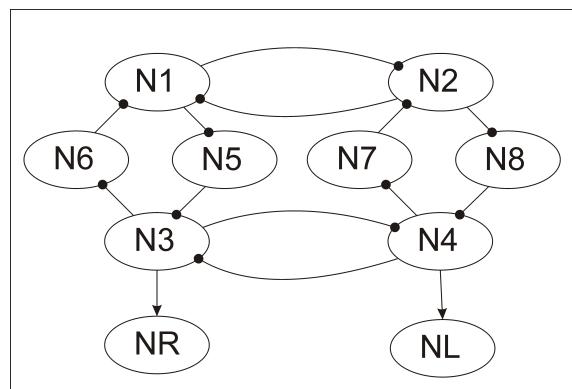


FIGURE 3.14 – Réseau de neurones utilisé dans le projet Hyrène. Ce réseau est le CPG présenté jusqu’ici, mais auquel nous avons ajouté une synapse excitatrice allant du neurone N₃ au neurone NR et une synapse excitatrice du neurone N₄ au neurone NL. Pour rappel, les neurones moteurs N₃/N₄ ont une activité en anti-phase. Ces 2 synapses sont plastiques et permettent de stimuler fortement les neurones NR/NL uniquement sur le premier PA de chaque bouffée de notre CPG.

En rappelant que notre stimulateur a deux voies d’entrées numériques, les PAs de NR sont injectés dans une première voie numérique permettant de déclencher une micro-stimulation visant

Paramètres des neurones NR et NL et des synapses les connectant au CPG									
a	b	c	d	I_{stat}	w	p	τ_{syn}	τ_{AMPA}	
0.02	0.2	-65	8	5.5	20	0.1	5 000 ms	10 ms	
							2 500 ms		
							0.9 1 250 ms		

TABLE 3.9 – Paramètres décrivant les neurones NR/NL et les synapses qui relient les neurones N3/N4. Ces paramètres sont classés dans l'ordre pour les CPGs de périodes moyennes de 10, 3 et 1.3 secondes.

les racines L2L/L5R tandis que les PAs de NL sont injectés dans la deuxième voie numérique pour déclencher une micro-stimulation visant les racines L2R/L5L.

La Figure 3.15 nous montre un exemple où les neurones NR/NL émettent des PAs alternés d'une période moyenne de 10 secondes. Toujours en suivant la course d'action présentée dans la Figure 3.13, ces PAs vont être injectés dans le stimulateur qui va, à son tour, provoquer une micro-stimulation dans les racines nommées précédemment. Nous constatons alors que les racines rendues inactives par la lésion de la moelle épinière reprennent une activité en bouffée cadencée par les PAs des neurones NR/NL.

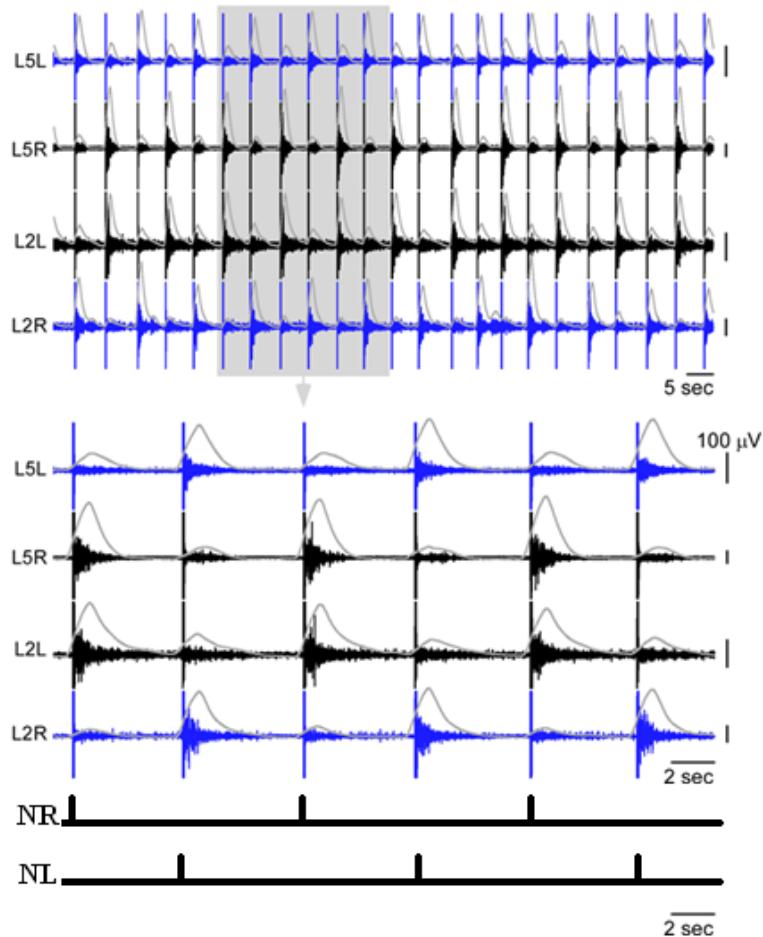


FIGURE 3.15 – Enregistrements de l'activité des racines ventrales L2/L5 suite aux stimulations pilotées par les PAs des neurones NR et NL.

Cet excellent résultat nous a montré avec succès la possibilité de stimuler le vivant en pilotant des micro-stimulations par l'activité d'un CPG numérique. Par la même occasion, nous avons montré une preuve de concept d'une future génération de neuroprothèse neuromorphique. L'ajout d'un réseau de neurones dans une neuroprothèse permet d'amener de la flexibilité dans les stimulations des tissus biologiques.

3.4 Conclusion et Perspectives

Le Chapitre 3 nous a permis de mettre en action les différentes options du réseau de neurones et nous montre des exemples du potentiel de notre réseau de neurones comme :

- la plasticité à court-terme et l'effet GABA-ergique qui permettent d'émuler un comportement similaire aux CPGs.
- le délai synaptique qui décrit la distance entre les neurones.
- le bruit synaptique qui permet de générer dans notre réseau de neurones une activité spontanée similaire à celle observée dans un réseau de neurone biologique.

Au final, l'implémentation et la configuration de notre réseau de neurones ont été un succès dans les projets BRAINBOW et HYRENE. L'utilisation de la plasticité à court-terme et de l'effet GABA-ergique au sein de notre réseau de neurones sont des éléments clés qui permettent d'émuler le comportement des CPGs. L'implémentation des CPGs sur une plateforme numérique (Multimed) est la partie de ce travail de thèse pour le projet HYRENE. Par contre, toutes les options présentées dans le Chapitre 2 et le Chapitre 3, à savoir la plasticité à court-terme, le bruit et le délai synaptique, ont été regroupées dans le projet BRAINBOW. Le projet BRAINBOW se distingue d'autres travaux sur les neuroprothèses (Prueckl et al., 2011)(Bamford et al., 2012)(Berger et al., 2012)(Hampson et al., 2012)(Opris et al., 2012), par l'exploitation de l'électrophysiologie in vitro, caractérisée notamment par la visibilité et le contrôle des conditions physiques et chimiques (Bonifazi et al., 2013). Ces atouts permettent d'étudier le traitement de l'information dans un réseau de neurones et permettent de comprendre comment interfaçer l'artificiel (dont notre réseau de neurones numérique fait partie) avec le vivant. Les options de notre réseau de neurones nous ont permis d'établir un pont entre deux clusters isolés (Bonifazi et al., 2013).

Le projet HYRENE présente une nouvelle optique pour réhabiliter le vivant (moelle épinière lésée) par l'utilisation d'un CPG artificiel (Joucla et al., 2013).

Gen 2 :8-14, " Puis l'Eternel Dieu planta un jardin en Eden, du côté de l'orient, et il y mit l'homme qu'il avait formé. L'Eternel Dieu fit pousser du sol des arbres de toute espèce, agréables à voir et bons à manger, et l'arbre de la vie au milieu du jardin, et l'arbre de la connaissance du bien et du mal.

Un fleuve sortait d'Eden pour arroser le jardin, et de là il se divisait en quatre bras.

Le nom du premier est Pischon ; c'est celui qui entoure tout le pays de Havila, où se trouve l'or. L'or de ce pays est pur ; on y trouve aussi le bdellium et la pierre d'onyx.

Le nom du second fleuve est Gihon ; c'est celui qui entoure tout le pays de Cusch.

Le nom du troisième est Hiddékel ; c'est celui qui coule à l'orient de l'Assyrie.

*Le quatrième fleuve, c'est l'Euphrate.
"*

CHAPITRE 4

VERS UN PROTOCOLE UNIVERSEL DE COMMUNICATION POUR SYSTÈMES NEUROMORPHIQUES

L'HYBRIDATION de réseaux de neurones est envisageable à différentes échelles. Cela peut aller de la conception d'une neuroprothèse (échelle microscopique de l'hybridation) à l'étude de la dynamique de grands réseaux. Pour cette dernière, il faut alors envisager d'interconnecter plusieurs plateformes neuromorphiques, voire même des plateformes de différents types. Ces types sont :

- informatique (dans le sens d'une implémentation logicielle). Un ordinateur ou un logiciel peut aisément simuler un réseau de neurones et ainsi générer des potentiels d'action (PAs) ;
- électronique (dans le sens d'une implémentation matérielle). Un ensemble de circuits électroniques (analogique et/ou numérique) sont à l'origine des PAs émis ;
- biologique. Il s'agit dans notre cas de neurones en culture sur des électrodes.

Les plateformes neuromorphiques ne contiennent qu'un nombre limité de neurones (de 100 à plusieurs millions selon notre état de l'art). Cependant, augmenter la taille des réseaux de neurones permettrait de mener d'autres études sur le cerveau, par exemple. Il est donc devenu crucial de faire entrer en communication les plateformes neuromorphiques afin d'obtenir un réseau plus grand.

Dans ce chapitre, nous allons présenter un protocole très connu au sein de la communauté des concepteurs neuromorphiques ainsi que ses limites. Ce protocole s'appelle le protocole AER (pour **A**ddress **E**vent **R**epresentation). L'idée d'origine du protocole AER est de pouvoir échanger entre 2 différentes plateformes neuromorphiques d'un même type, les différents potentiels d'actions qu'elles émettent. Lors de cette présentation, nous allons aussi constater que ce protocole a été adapté suivant les divers besoins des implémentations neuromorphiques voulant mettre en œuvre le protocole AER. Ces différentes versions demeurent incompatibles entre elles, et posent une limitation à une très large utilisation des plateformes neuromorphiques.

Pour résoudre cette problématique, l'idée serait de créer alors un protocole simple, permettant par l'intermédiaire d'un média commun, d'échanger librement les informations importantes. Nous choisissons cette fois-ci de parler d'informations importantes car nous avons vu précédemment que les PAs ne sont pas les seules informations utiles et détectées au sein d'un réseau de neurones (par exemple, il y a aussi la présence d'une bouffée de PAs ou de LFP (**L**ocal **F**ield **P**otential)).

Viser une hybridation plus large signifierait donc d'être capable d'abolir la frontière entre l'électronique, l'informatique et le biologique afin de créer un immense réseau de réseaux neuromorphiques. Après avoir présenté un réseau de neurones numérique pouvant interagir avec le vivant (Chapitre 2 et Chapitre 3), ce chapitre propose le protocole GHON ((**G**eneric **I**nterconnection between **H**eterogeneous spiking neurONs)) comme solution à cette problématique, en utilisant l'Ethernet comme média commun pour les différents échanges. Nous verrons comment, au travers de l'Ethernet, il est possible de faire communiquer ces 3 types de réseaux de neurones.

4.1 Le protocole AER

4.1.1 Introduction

Dans un réseau de neurones, les potentiels d'action sont l'essentiel de l'information nerveuse, échangés entre un neurone pré-synaptique et ses neurones post-synaptiques. Le protocole doit être capable de faire la correspondance entre les neurones.

Le protocole AER (Adress-Event Representation) se propose, au travers d'un bloc de mapping (appelé «mappeur», connaissant l'interconnexion entre les neurones d'une plateforme vers les neurones d'une autre plateforme) et de bus de données, de convoyer un potentiel d'action de l'émetteur vers son ou ses destinataires. Plus précisément, à l'émission d'un potentiel, le neurone pré-synaptique va transmettre au mappeur son identifiant.

Grâce à la table de connectivité stockée dans le mappeur, dès la réception de l'identifiant du neurone pré-synaptique (neurone source), le mappeur enverra en série l'identifiant de chaque neurone post-synaptique (neurone cible). Chaque neurone ainsi identifié reçoit le potentiel d'action.

Le protocole AER permet de manière asynchrone de convoyer les potentiels d'action d'une plateforme à une autre. A l'origine, ce protocole utilise un mappeur global et une série de bus, de multiplexeurs et de démultiplexeurs. L'inconvénient de ce protocole est qu'il a été modifié pour en augmenter les performances et aussi pour l'adapter aux différents montages neuromorphiques. (Zamarreno-Ramos et al., 2013) fait une revue des différents protocoles AER ainsi créés et nous présente 5 versions les plus communes. Cependant ces 5 protocoles, bien que basés sur le même principe, n'en demeurent pas moins incompatibles les uns avec les autres.

AER simple

Chaque plateforme possède une entrée AER_in et une sortie AER_out. Chaque fois qu'un neurone d'une plateforme émet un potentiel d'action, alors la plateforme transmet l'identifiant du neurone (i.e numéro de neurone). Un arbitre reçoit toutes les requêtes ainsi émises les ordonnant avant de les transmettre au mappeur. Dans le protocole AER, toutes les plateformes voulant communiquer sont classées selon un ordre de priorité. Cette propriété fait du protocole AER un protocole non égalitaire.

Une fois que le mappeur reçoit un identifiant, il associe immédiatement l'identifiant du neurone émetteur, ainsi que le numéro de la plateforme dont il vient, aux neurones post-synaptiques. Le mappeur enverra en série (l'un à la suite de l'autre), un message à chaque neurone post-synaptique. Ici, le mappeur est alors appelé mappeur global, car il est le seul bloc à connaître les connexions inter-plateformes (Figure 4.1). Evidemment, en reprogrammant le mappeur, il est possible de changer la connectivité inter-plateforme à volonté. Nous prendrons par convention une flèche colorée pour représenter les bus de données AER.

En adoptant une vue macroscopique, ce protocole est un lien de communication point à point (c'est à dire un neurone pré-synaptique (émetteur) aura une liste de neurones selon laquelle chaque neurone post-synaptique (destinataire) sera atteint en série), pondéré par un mappeur global (Mortara et al., 1995)(Boahen, 2000)(Boahen, 2004)(Chicca et al., 2007)(Vogelstein et al., 2007)(Mayr et al., 2005)(Cassidy et al., 2011).

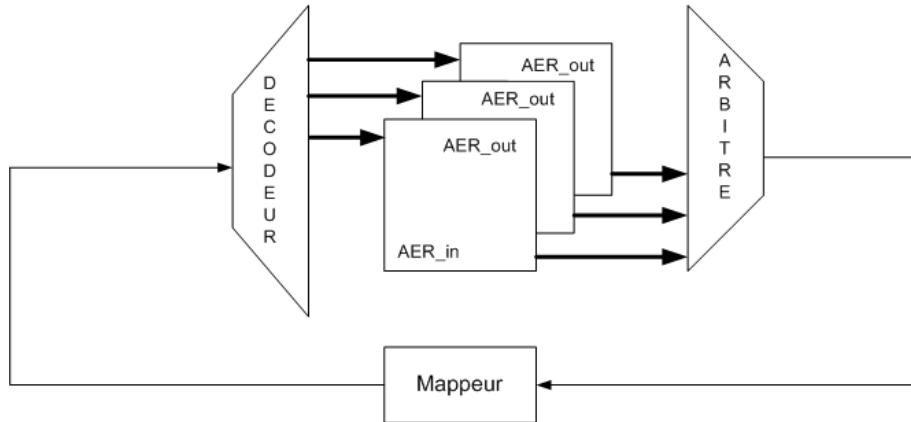


FIGURE 4.1 – Représentation de l’AER simple.

AER à nœuds de diffusion

Dans ce protocole représenté dans la Figure 4.2, chaque plateforme possède deux voies d'accès, la première allant des entrées AER_i vers les sorties AER_i' (voie descendante) et la seconde allant des entrées AER_o vers les sorties AER_o' (voie ascendante).

De cette manière, à chaque fois qu'un neurone d'une plateforme émet un potentiel d'action, son identifiant (numéro de plateforme et/ou numéro de neurone) est transmis en utilisant la voie ascendante pour atteindre le mappeur. Ce dernier va diffuser l'événement à toutes les plateformes en utilisant cette fois-ci la voie descendante.

Pour finir, si une information entrant par AER_i d'une plateforme concerne son réseau de neurones, alors elle la lui transmet, sinon l'information est simplement recopiée sur la sortie AER_i' afin d'atteindre la plateforme suivante.

Ce protocole est donc un lien de communication point à point multiple, agissant avec l'aide d'un mappeur global (Lin et al., 2006)(Merolla et al., 2007)(Bamford et al., 2010).

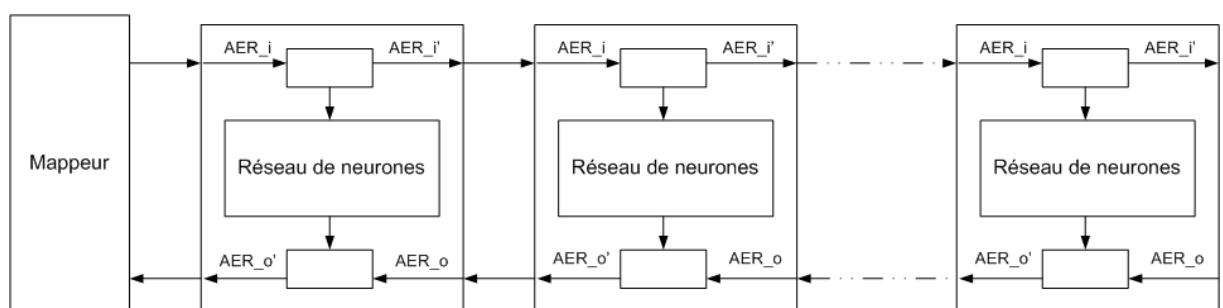


FIGURE 4.2 – Représentation de l’AER à nœuds de diffusion.

AER pré-structuré

Ce type de protocole (voir Figure 4.3) s'affranchit de la présence d'un mappeur global en le substituant par un ensemble de nœuds de centralisation, notés "M" dans la figure pour *Merger*, de répartition, notés "S" pour *Splitter*, et aussi de mappeurs locaux selon les versions. Ces différents nœuds forment une structure qui doit être définie en même temps que l'architecture du réseau.

L'optimisation de cette conception permet d'éviter de transporter inutilement les événements sur toutes les plateformes.

Le bloc S possède une entrée et plusieurs sorties, ce qui permet de partager des informations provenant d'une plateforme vers plusieurs autres plateformes. Le bloc M possède plusieurs entrées et une sortie, ce qui permet de rassembler des informations provenant de plusieurs plateformes vers une seule plateforme.

Même si cette catégorie d'AER augmente l'efficacité des transports d'événements, elle rend par contre plus difficile la reconfigurabilité. En effet, pour changer la connectivité entre 2 plateformes, il devient nécessaire de changer manuellement les nœuds qui les relient ainsi que d'ajouter/enlever les bus de données qui font le pont entre les nœuds et les plateformes.

Ce protocole est un protocole point à point qui met en scène des mappeurs locaux à défaut d'utiliser un mappeur global (Serrano-gotarredona et al., 2009).

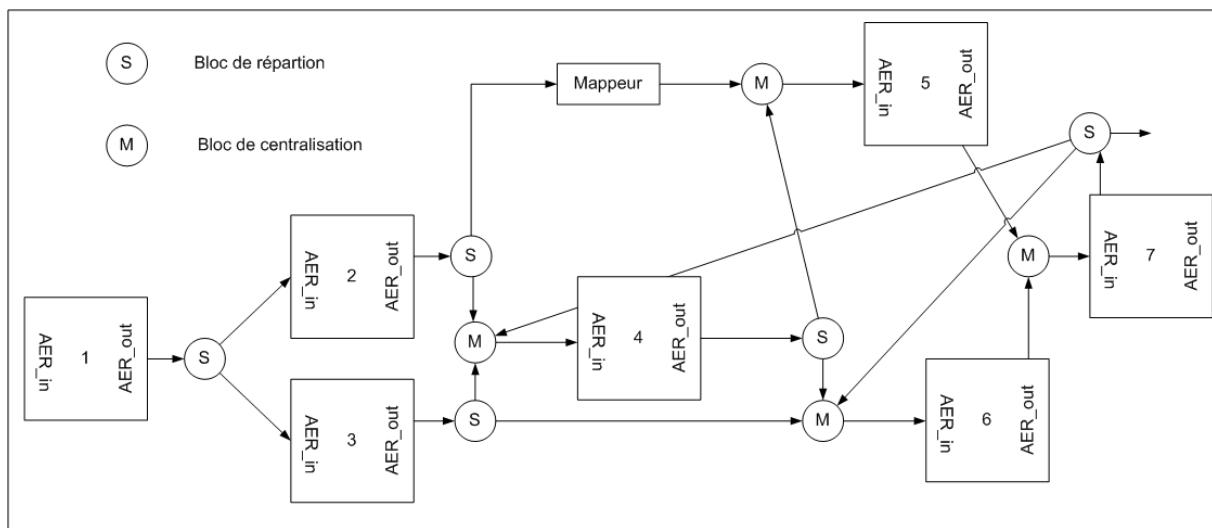


FIGURE 4.3 – Une représentation de l'AER pré-structuré.

AER à fractale hiérarchique

Nous emploierons le mot « macro-tuile » pour désigner l'ensemble décrit par la Figure 4.4 et « tuile intérieure » pour chaque ensemble décrit par un carré contenant des flèches jaunes. De la même manière, nous appellerons « micro-tuile » tout élément représenté par un carré contenant les mots AER_in et AER_out.

Cette hiérarchisation a donné son nom à ce type de protocole car chaque macro-tuile est elle-même composée de tuiles intérieures, qui sont à leur tour composées de micro-tuiles. Ce protocole est l'extension de l'AER simple basé sur l'hypothèse que les neurones proches (neurones des micro-tuiles) sont plus densément connectés que les neurones plus éloignés (neurones d'une des tuiles intérieures ou même les neurones d'une autre macro-tuile).

Elle favorise la rapidité entre les micro-tuiles avec des bus parallèles, tandis la communication entre les macro-tuiles est faite en série. Dans cette version du protocole AER, chaque mappeur contient la connectivité interne (i.e la connectivité inter-tuile) et la connectivité externe (connectivité inter-macro-tuile).

Ce protocole est la dernière version point à point que nous verrons, utilisée par (Joshi et al., 2010).

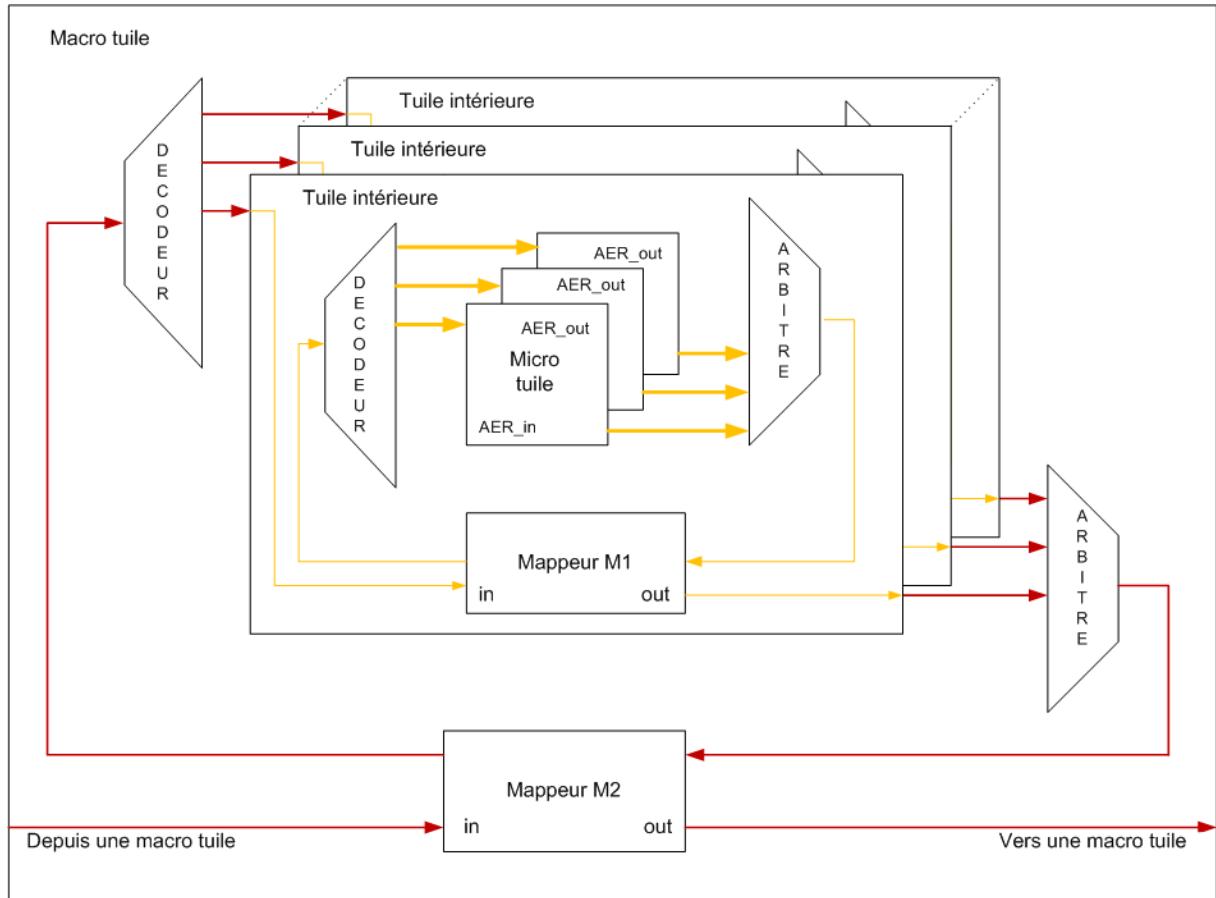


FIGURE 4.4 – Représentation de l'AER à fractal hiérarchique.

AER à mailles de routage

Les versions du protocole AER présentées jusqu'à maintenant sont à une dimension. L'AER a été aussi adapté en une version 2D, dans laquelle chaque réseau de neurones est empaqueté avec un routeur local formant ainsi un module (représenté ici par un carré jaune dans la Figure 4.5). Cette structure est utilisée notamment dans la plateforme Spinnaker (Furber et al., 2013). En utilisant un jeu d'instructions précis, lors de l'émission d'un potentiel d'action, le routeur le plus proche va transmettre l'événement à l'intéressé (ou aux intéressés) de manière optimale, en joignant le minimum de modules.

Une variante de cette version existe et porte le nom d'AER à maille multicast, qui est très proche de cette version. Le terme multicast implique que l'envoi d'un message ou d'une donnée n'est effectué qu'à un nombre restreint d'adresses (ici, un nombre restreint de modules) (Khan et al., 2008).

4.1.2 Synthèse sur le protocole AER

Le protocole AER est un protocole asynchrone, non égalitaire et point à point, qui repose sur le concept d'identification, soit de l'émetteur d'un potentiel d'action, soit des destinataires du potentiel d'action.

Mais ces différentes déclinaisons sont peu compatibles (voire même incompatibles) les unes avec les autres.

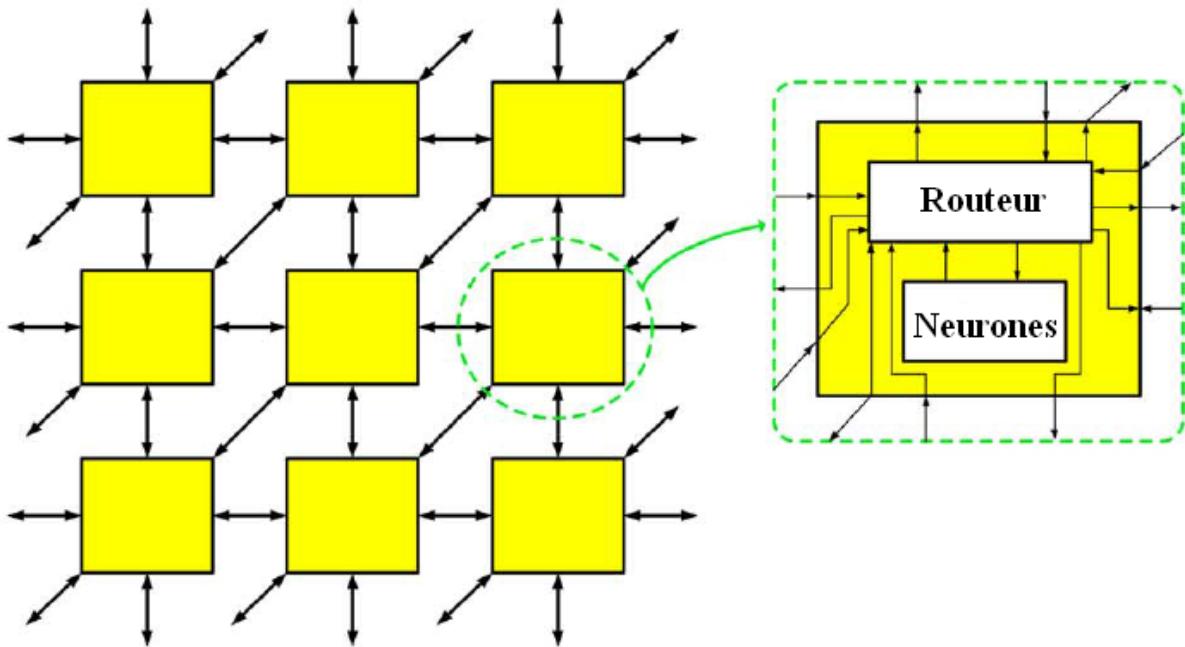


FIGURE 4.5 – Représentation de l'AER à mailles routeurs.

Pour illustrer ce propos, nous pouvons nous reporter aux travaux de (Rast et al., 2013) qui présentent la communication entre 2 plateformes neuromorphiques utilisant la version de l'AER à mailles de routage . L'une de ces plateformes est celle développée par l'Université d'Heidelberg durant le projet BrainScaleS et la seconde est SpiNNaker de l'Université de Manchester. Les deux plateformes utilisent pour leur communication interne un AER mais le contenu des trames asynchrones est différent. Pour que ces deux plateformes puissent communiquer, il est nécessaire d'ajouter sur chaque plateforme un module de conversion adaptant les trames de la plateforme émettrice pour les rendre compatibles avec la plateforme réceptrice.

Pour surpasser ces limitations, il faudrait créer un protocole de communication capable de connecter entre-elles des plateformes neuromorphiques électroniques mais aussi informatiques ou biologiques. Ceci est le but du protocole GIHON (Generic Interconnection between Heterogeneous spiking neurONs).

Le protocole GIHON utilisera un bus Ethernet commun avec lequel il sera possible d'envoyer des messages-événements (i.e potentiel d'action) d'une plateforme à une autre. Dans la suite de ce chapitre, nous discuterons plus précisément des spécifications du protocole universel, open-source GIHON.

4.2 Particularité du protocole UDP

Le protocole UDP (pour User Datagram Protocol) fait partie de la suite de protocole internet (ou aussi appelée la suite TCP/IP) dans laquelle se trouvent le protocole TCP et le protocole IP (Internet Protocol) et fait partie de la couche 4 du modèle OSI (couche Transport).

Le rôle de ce protocole est de permettre la transmission de données de manière très simple entre deux entités, chacune étant définie par une adresse IP et un numéro de port. Contrairement

au protocole TCP, il fonctionne sans négociation : il n'existe pas de procédure de connexion préalable à l'envoi des données.

4.2.1 Notre intérêt autour du protocole UDP

Nous avons vu dans la Section 4.1.2, le protocole GIHON permettra de transporter les informations d'une plateforme vers les autres.

Le protocole UDP a été choisi pour transporter les trames GIHON car il permet de remplir cette tâche. Les principaux avantages de ce protocole est qu'il est exempté de négociations (ou *handshaking*, procédure selon laquelle on vérifie que le destinataire d'un message est bien connecté et aussi a bien reçu le message) et la vérification de checksum demeure facultative. Ces avantages permettent d'alléger l'implémentation et la rende ainsi plus rapide. En effet, l'UDP ne se charge que du transport de données sans se soucier de la validité de son contenu. Il appartient donc au programme utilisateur de gérer cela.

De plus, le protocole UDP permet d'avoir plusieurs destinataires (multicast) et permet aussi de cibler tous les membres connectés (broadcast). Cette dernière propriété est le centre du protocole GIHON.

4.2.2 Paquets et encapsulations des données

(Hornig, 1984) définit comment procéder pour encapsuler les datagrammes IP sur un réseau Ethernet (et donc une méthode pour émettre des données via le protocole UDP, encapsulé dans le protocole IP (Protocol, 1980)).

Les trames de l'Ethernet sont composées de 5 parties. Par ordre d'apparition, nous avons l'adresse MAC destination (codée sur 6 octets), l'adresse MAC source (codée sur 6 octets), le type de protocole encapsulé dans cette trame (appelé "*ether type*" et codé sur 2 octets), les données à transmettre (appelés "*payload*" et sa taille varie entre 46 à 1500 octets) et le checksum CRC (codé sur 4 octets) permettant de vérifier la validité du message Ethernet.

Pour utiliser le protocole UDP, l'*"ether type"* doit désigner un protocole IP. De cette manière, nous indiquons que la suite de la trame (qui est le contenu du "*payload*") appartient au protocole IP. Le "*payload*" sera alors divisé en 3 parties qui sont, dans l'ordre, l'en-tête IP (codé sur 20 octets), l'en-tête UDP (codé sur 8 octets) et les données à transmettre.

L'en-tête IP donne des renseignements sur la trame comme sa taille, sa durée de vie et surtout le protocole qu'elle encapsule (dans notre cas le protocole UDP).

L'en-tête UDP est composé des adresses IP de la plateforme-source (plateforme du neurone pré-synaptique de potentiel d'action) et de la plateforme-destination (plateforme du groupe de neurones post-synaptiques), la longueur des informations et un checksum sur 16 bits chacun.

4.3 Le protocole GIHON

4.3.1 Cahier des charges du protocole GIHON

Le protocole GIHON est un protocole UDP standard qui est compatible avec le temps des systèmes neuromorphiques. Les paquets UDP générés dans le protocole GIHON auront des en-têtes de 42 octets.

Le protocole GIHON sera mis en action via une connexion Ethernet à 100 Mb/s utilisant le protocole UDP. Ce protocole devra être capable de transporter plusieurs types d'information sous forme de paquets de 32 bits. Parmi les paquets à transmettre, nous avons :

- un paquet d'identification contenant l'identifiant du neurone ayant émis un potentiel d'action. L'un des intérêts d'utiliser le protocole UDP est la possibilité d'identifier la plate-forme au sein de laquelle un neurone aurait émis un PA grâce l'adresse IP. Pour permettre l'identification du neurone émetteur, il est nécessaire d'ajouter un paquet d'identification. Avec 24 bits, le paquet d'identification permet de distinguer jusqu'à 16 777 216 neurones (dans le Chapitre 1, nous avons présenté une plateforme possédant 1 million de neurones ([Cassidy et al., 2011](#))).
- un paquet d'horodatage permettant de situer temporellement l'occurrence d'un PA émis. La plus petite valeur d'horodatage représentée sera alors $50 \mu\text{s}$. Cette valeur a été choisie car elle correspond au plus petit temps de réponse dans la discrimination visuelle ([Benosman et al., 2012](#)). Ainsi avec un horodatage codé sur 24 bits, nous obtenons alors une plage roulante de 14 minutes. En gardant une trace des roulements, il sera possible d'identifier n'importe quel PA apparaissant dans une expérience raisonnable (où le temps maximal séparant 2 PA consécutifs d'une même source est inférieur à 14 minutes). Afin que l'horodatage soit pertinent, il est impératif que tous les participants d'une même conversation soient synchronisés temporellement grâce à une machine d'états commune.

4.3.2 Anticipation de l'avenir

Cette partie du travail de thèse ne représente que le début du protocole GIHON. Celui-ci est amené à évoluer et -nous l'espérons- être utilisé au sein de la communauté neuromorphique. C'est pourquoi dès la conception du protocole GIHON, nous avons choisi d'ajouter 8 bits en tête des paquets de 24 bits d'information (à ce jour utilisé pour l'identification ou l'horodatage), permettant ainsi de distinguer le type d'information transportée.

Dans un futur proche, les 8 bits réservés pourront alors décrire un autre type d'émetteur (comme une population de neurone), ou une amplitude de potentiel d'action ou la présence d'un burst par exemple.

Par conséquent, le payload d'une trame GIHON sera composée de segments de 32 bits. A ce jour, le payload d'une trame GIHON sera composé de 2 segments. Le premier segment a ses 8 bits de poids forts réservés suivi de 24 bits codant l'identifiant du neurone ayant émis un PA. Avec une répartition similaire, le second segment est découpé en 8 bits de poids forts réservés et 24 bits de poids faibles servant à l'horodatage du PA émis.

Grâce à ces dispositions, une connexion Ethernet à 100 Mbits/s peut supporter 260 000 événements/s pour ces paquets de 48 octets (atteignable avec un UDP core sur un FPGA).

En suivant un protocole UDP standard, les paquets UDP d'origine informatique (générés par un langage de programmation) peuvent interagir avec d'autres systèmes neuromorphiques. En effet, un code C++ optimisé peut générer entre 20 000 et 40 000 paquets UDP par seconde.

Par ailleurs, il serait possible de renvoyer plusieurs séquences de potentiels d'actions stockés dans un disque dur pourvoyant des stimuli enregistrés au préalable via UDP.

4.3.3 Architecture sur le FPGA

L'architecture se répartit sur 3 étages (Figure 4.6) :

- un microprocesseur (ici Microblaze) qui se chargera de la communication Ethernet.
- un étage intermédiaire qui se charge d'une part de lister à partir du vecteur de PA, l'ID de chaque neurone ayant fourni un PA et d'autre part d'accumuler les ID réceptionnés et transmettre un vecteur de PA au réseau de neurones numérique.

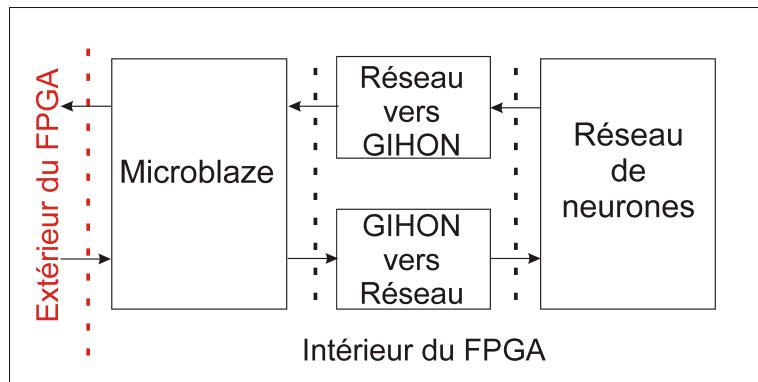


FIGURE 4.6 – Architecture mettant en œuvre GIHON sur FPGA. L’architecture est divisé en 3 parties : un microprocesseur Microblaze, une couche intermédiaire de relais (les blocs "GIHON vers réseau" et "Réseau vers GIHON") et le réseau de neurones numériques.

- un réseau de neurones. Dans ce chapitre, nous mettrons en action le réseau de neurones présenté dans le chapitre 2.

Le microprocesseur Microblaze

Le MicroBlaze est un cœur de processeur softcore 32 bits créé par la société Xilinx, conçu pour être implanté sur les FPGA appartenant à ce fabricant.

Le cœur de processeur est fourni avec un ensemble de bibliothèques C permettant de gérer les différentes entrées/sorties qui lui sont associées à l'aide de macros.

Pour configurer le microprocesseur Microblaze, nous précompilons un code C. Lors de la synthèse finale, ce code C compilé sera réutilisé pour compléter le design VHDL de l'architecture globale. L'architecture proposée est similaire à tous les intervenants.

L'utilisation d'un microprocesseur rend cette architecture plus flexible et plus universel. La souplesse provient du code C, qui nous permet de changer le comportement global du microprocesseur au besoin sans passer par de longues phases de synthèse. De l'autre côté, l'universalité provient du code VHDL en lui-même qui est commun à toutes les plateformes FPGA. Cependant, le microprocesseur varie suivant le constructeur. Par exemple, Altera a le processeur Nios II.

Dans notre cas, le microprocesseur MicroBlaze abrite le protocole GIHON et prend en charge les entrées/sorties Ethernet sur notre plateforme FPGA. Le protocole GIHON exige un identifiant du neurone (numéro du neurone ayant émis un PA) et le temps auquel ce PA a été émis. Sur la base de ces 2 informations, le microprocesseur MicroBlaze émettra une trame UDP dont les données seront en premier l'identifiant du neurone suivi du temps d'émission du PA.

La couche intermédiaire

La couche intermédiaire est une couche stratégique qui s'adapte au réseau de neurone en amont. Cette couche a pour effet d'une part de convertir les paquets GIHON en un vecteur de PA utilisable pour notre réseau de neurones (bloc "GIHON vers Réseau") et d'autre part de récupérer les identifiants des neurones ayant émis un PA et l'horodatage qui leur correspond (bloc Réseau vers GIHON").

Dans notre cas, les PAs sont émis de manière synchrone et sont rassemblés en un vecteur de PA. C'est ainsi que le bloc "Réseau vers GIHON" exploitera le vecteur de PAs et émettra en série les identifiants des neurones ayant émis un PA.

Le réseau de neurones

En amont de la connexion Ethernet se trouve le réseau de neurones. Ce bloc abrite aussi un compteur (sur 23 bits cadencé à $50 \mu\text{s}$) permettant d'horodater l'émission des PAs.

Nous utiliserons ici le réseau de neurones présenté au Chapitre 2.

Synthèse du protocole GIHON

Un protocole est avant tout un concept, un ensemble de règles contrôlant un échange d'informations entre plusieurs intervenant au travers d'un média commun. Dans le protocole GIHON, les intervenants s'échangent les identifiants des PAs et leurs horodatages au travers du réseau Ethernet.

Selon ce protocole, à chaque fois qu'un intervenant abrite un neurone émettant un PA, cet intervenant doit émettre alors une trame GIHON (Figure 4.7) selon les conditions spécifiées par le protocole Ethernet.

Par analogie aux matriochkas, le protocole GIHON est un protocole (poupée supérieure), qui encapsule 3 autres protocoles (poupées intérieures). En effet, le protocole GIHON encapsule (par ordre d'importance) le protocole Ethernet II qui englobe le protocole IPv4 qui lui-même contient le protocole UDP.

C'est ainsi qu'une trame GIHON est alors composée successivement d'un en-tête MAC (Adresse MAC destination puis l'adresse MAC source), suivi d'un en-tête IP, puis d'un en-tête UDP, se terminant sur les données à transmettre (1 octet réservé + 3 octets codant l'identifiant du neurone émettant un PA et 1 octet réservé + 3 octets codant l'horodatage du PA émis).

Pour la partie technique, la trame GIHON est une trame ETHERNET II (UDP) dans laquelle les données transmises sont à ce jour l'identifiant des neurones puis l'horodatage. Etant donné que tous les intervenants dans le protocole GIHON sont connectés au réseau Ethernet, l'adresse MAC destination sera toujours l'adresse Broadcast (i.e 255.255.255.255.255.255). En effet, cette adresse permet ainsi de cibler tous les intervenants connectés.

L'adresse MAC source est bien sûr l'adresse MAC de l'intervenant émetteur. Les 2 octets correspondant au type d'IP est 0x0800 pour l'IPv4.

Il appartient ensuite à chaque intervenant de traiter les données transmises et aussi de se rendre apte à utiliser ce protocole. Ces 2 points rendent la couche intermédiaire flexible suivant le réseau de neurone en amont.

4.3.4 Machine d'état gérant la synchronisation et l'envoi des données

Tous les intervenants neuromorphiques doivent impérativement posséder le protocole GIHON qui agira à la manière d'un langage universel. Le protocole met en œuvre un système de maître & esclave. Le maître au sein de ce réseau de réseau va démarrer une phase d'initialisation. Cette fonction prendra fin au moment où il enverra un top départ à tous les esclaves déjà initialisés. Chaque esclave est en attente d'une discussion d'initialisation avec le maître et dès lors qu'ils deviennent initialisés, ils demeurent en attente du top départ donné par le maître.

Cette discussion commence par le maître sur la plateforme P1, étant initialisé dans l'état CALL tandis que les esclaves P2 et P3 débutent dans l'état ANSWER . Dès le départ, le maître ne

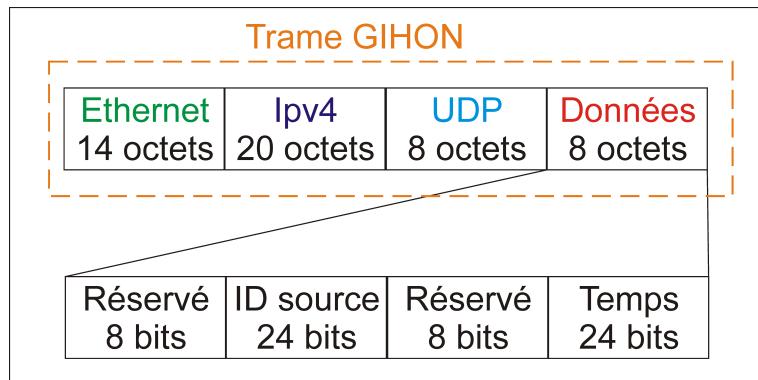


FIGURE 4.7 – Présentation d'une trame GIHON. Comme toute trame Ethernet II (UDP), elle démarre avec l'adresse MAC destination (Dans GIHON, cette adresse est l'adresse Broadcast).

connaît pas le nombre d'esclave présents et connectés à l'Ethernet. Le maître envoie un message UDP à tous les possibles intervenants puis passe en attente de réponse (état REGISTER).

Mettons que P2 soit le premier à répondre alors P1 enregistre l'IP de P2. L'état intermédiaire ARP & UDP permet de rendre le maître indisponible pour tout autre intervenant que P2. Dans cet état, P1 envoie une trame UDP à P2 et déclenche un chronomètre afin de mesurer le temps de réponse de P2.

Comme nous montre la Figure 4.3.4, P2 est dans l'état ANSWER_PING dans lequel à la réception d'une trame UDP provenant du maître (ici P1) va lui retourner une trame UDP en réponse. Une fois la réponse adressée, P2 passe alors à l'état SYNCHED.

A la réception de la réponse de P2, P1 passe en état STOP_CHRONO qui lui permet de stopper le chronomètre précédemment lancé. P1 récupère alors la valeur du chronomètre, calcule le temps de mi-parcours (état 7) puis renvoie à P2 la valeur ainsi calculée (état 8). Une fois ces étapes complétées, P1 passe en état 12 (CHECK_SYNCHED_VALUE).

P2 reçoit la valeur calculée et renvoie à P1 cette dernière (état 10) et passe alors en état 11 (WAIT_VALIDATION). P1 reçoit la valeur et vérifie qu'elle correspond à celle qu'il avait envoyé précédemment. Si cela est le cas alors il envoie une trame de confirmation à P2, qui passera dans l'état 15 (PAUSE). L'esclave P2 est alors considéré comme synchronisé (le temps de mi-parcours est la moitié du temps total).

Dans le cas où la valeur envoyée par P2 ne correspond pas à la valeur calculée par P1, P1 passe par l'état 13 de temporisation (TEMPO) avant de recommencer la procédure depuis l'état 4 (ARP & UDP).

Si par contre la valeur est correcte, P1 passe en état 14 où il vérifiera la liste des esclaves à cibler. Si la liste est vide, cela signifie qu'il ne reste plus d'esclave à synchroniser. Dans ce cas là, P1 passe dans l'état PAUSE durant lequel il va lancer une trame UDP qui a pour but de lancer P2 dans l'état 16 (COMPUTATION) et passe lui même dans l'état 16.

Dans l'état 16, toutes les plateformes neuromorphiques sont autorisées à générer des trames GIHON pour signaler l'activité des neurones qu'elles abritent.

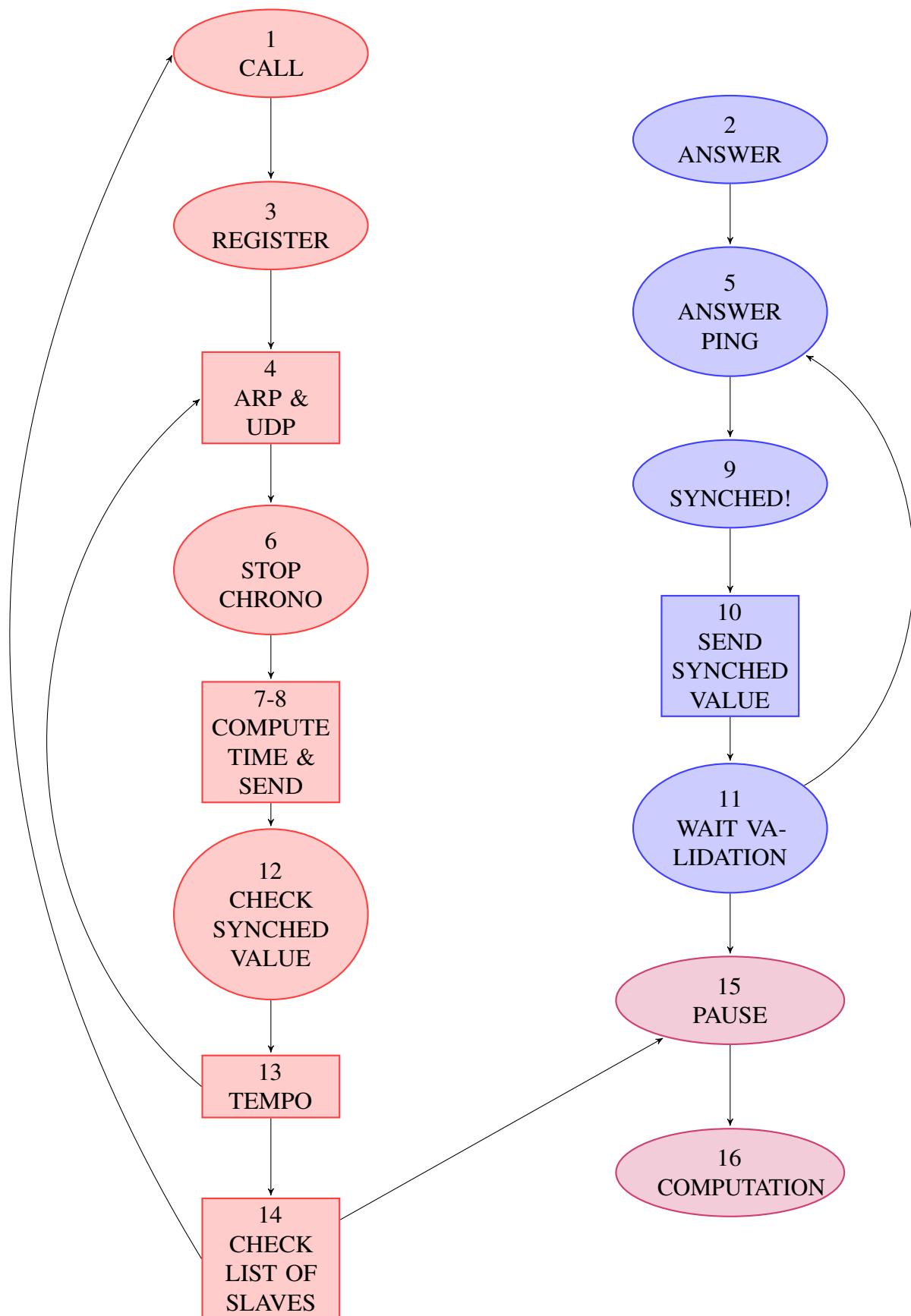


FIGURE 4.8 – Présentation de la machine d'états régissant les échanges du protocole GIHON.

4.4 Architectures pour réseaux de neurones informatique et biologique

Cette section est une discussion autour des possibilités offertes par le protocole GIHON. Ces différentes architectures permettent de mettre en action des réseaux de neurones informatiques ou biologiques. Chacune des architectures présentées auront un principe similaire à Figure 4.6 mais différeront dans l'origine du réseau de neurone et la mise en œuvre de sa couche intermédiaire.

4.4.1 Architecture comprenant un réseau de neurones logiciels

Le monde informatique nous donne 2 principales problématiques :

- la séquentialité du code. Dans le monde informatique, chaque ligne de code s'exécute les unes après les autres. Il est difficile de se repérer dans le temps.
- la vitesse de calcul des processeurs. La trop grande rapidité pose un problème pour simuler un petit réseau de neurones fonctionnant au temps biologique.

L'utilisation d'un timer peut réguler l'activité du réseau de neurones informatiques ainsi que l'envoi des trames GIHON.

D'un point de vue technique, le problème de la séquentialité peut être contourné en utilisant une ensemble de processus parallèles et d'interruptions (Figure 4.9).

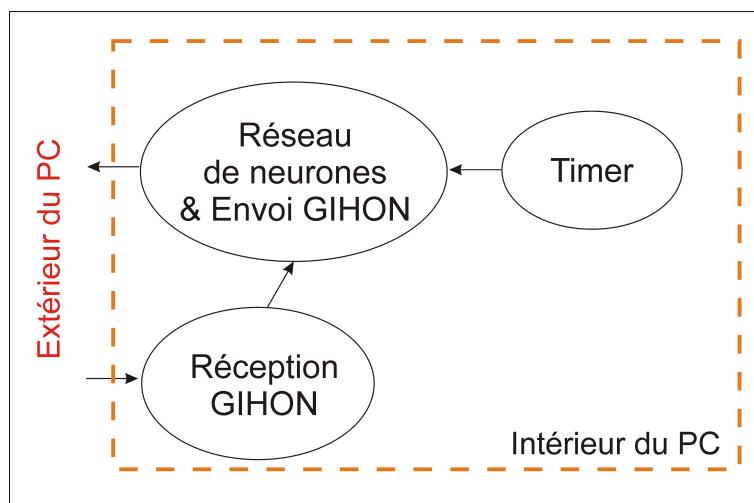


FIGURE 4.9 – Architecture mettant en œuvre GIHON avec un réseau de neurones informatiques. Les bulles représentent les processus parallèles et leurs rôles.

4.4.2 Architecture comprenant un réseau de neurones biologiques

Dans le chapitre 3, nous avons discuté du processus expérimental dans les projets BRAIN-BOW et HYRENE permettant entre autre de faire communiquer un réseau de neurones artificiels numériques avec un réseau de neurones biologiques. Cette communication n'avait pas pu se faire à l'aide de GIHON car la carte MultiMED où été implémenté le réseau de neurones est dépourvue de port Ethernet.

La Figure 4.10 décrit une architecture pouvant mettre en œuvre le protocole GIHON connecté à un réseau de neurones biologique. Cette architecture comprend un microcontrôleur Microblaze (gérant le protocole GIHON), un stimulateur qui stimulera le réseau de neurones biologique et une couche intermédiaire (les blocs "Réseau vers GIHON" et "GIHON vers Réseau"). Le bloc "Réseau vers GIHON" contient une chaîne de détection des PA_s biologiques, comme déjà implémentée dans les projets HYRENE et BRAINBOW, et permet de les transmettre au microcontrôleur qui les enverra au travers du protocole GIHON. Le bloc "GIHON vers Réseau" permet de traduire les PA_s transmis par GIHON en signaux ordonnant une stimulation comme cela avait été aussi déjà fait dans les deux projets.

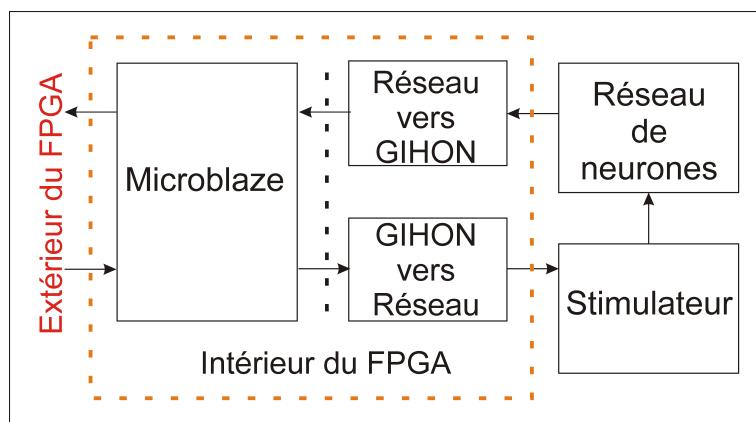


FIGURE 4.10 – Architecture mettant en œuvre GIHON avec un réseau de neurones biologiques. Les rectangles représentent les différents blocs nécessaires pour mettre en œuvre du protocole GIHON.

4.4.3 Synthèse sur les architectures mettant en œuvre le protocole GIHON

Les architectures présentées sont certes des suggestions d'architectures minimalistes, mais montrent qu'il est possible de connecter différentes plateformes neuromorphiques abritant différents type de réseaux de neurones grâce au réseau Ethernet et au protocole GIHON. La communication Ethernet met en relation plusieurs interlocuteurs, qui partagent un média commun (le réseau Ethernet). Pour avoir accès au réseau Ethernet, chaque interlocuteur doit être équipé d'un port Ethernet et la présence d'un routeur permet de connecter tous les interlocuteurs au même média.

L'expérience finale (non réalisée) consisterait à connecter ces différentes plateformes et obtenir une architecture semblable à la Figure 4.11.

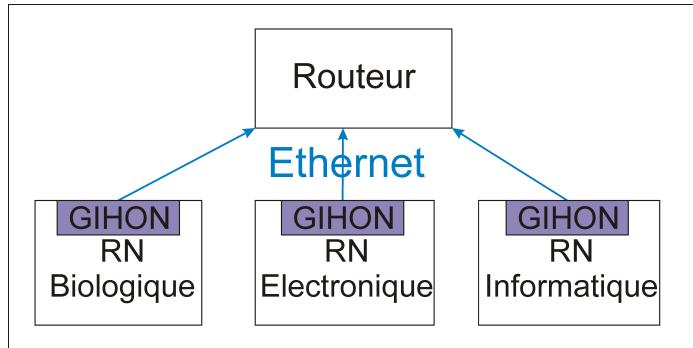


FIGURE 4.11 – Architecture mettant en œuvre l’architecture GIHON connectant un réseau de neurones électroniques, informatiques et biologiques. Chaque plateforme neuromorphique est relié à un routeur Ethernet.

4.5 Mise en œuvre du protocole GIHON pour deux plate-formes neuromorphiques

Ressources requises pour une implémentation

Nous avons implémenté l’architecture GIHON sur FPGA présenté dans la Figure 4.6 sur deux plateformes FPGA différentes (Plateforme 1 et Plateforme 2). La Table 4.1 nous montre les ressources nécessaires pour implémenter un RNA de 10 neurones et 46 synapses, un processeur Microblaze et les blocs de la couche intermédiaire entre le RNA et Microblaze.

Nous décidons de donner à la plateforme 1 l’adresse IP 169.254.151.133 et à la plateforme 2 l’adresse IP 169.45.166.12. Pour la suite de l’expérience, nous allons connecter les 2 plateformes et un ordinateur à un routeur Ethernet. De cette manière, les 2 plateformes peuvent communiquer librement en utilisant le protocole GIHON. L’ordinateur est présent sur le réseau afin de nous permettre de scruter les trames émises sur le réseau Ethernet en utilisant le logiciel Wireshark.

Ressources	Totales	Utilisées par MicroBlaze	Utilisées pour toute l’implémentation
Bascules	184 304	4 091	5 247
LUTs	92 152	4 358	6 098
Multiplieurs	180	3	39
RAM 16k	268	34	47

TABLE 4.1 – Liste des ressources occupées par Microblaze puis par l’ensemble de l’architecture présentée en Figure 4.6 comprenant un RNA de 10 neurones et 46 synapses.

Exemple de synchronisation GIHON

Nous trouvons en la Figure 4.12 un exemple qui illustre l’exécution de la machine à états présentée dans la Section 4.3.4. Dans la situation présentée, le maître sait que seul un esclave est connecté et va donc l’appeler (état 1) pour déclencher la synchronisation.

La machine à état régissant GIHON comporte des états intermédiaires (rectangles de la Figure 4.3.4) qui permettent de protéger et sécuriser la synchronisation et le dialogue entre 1

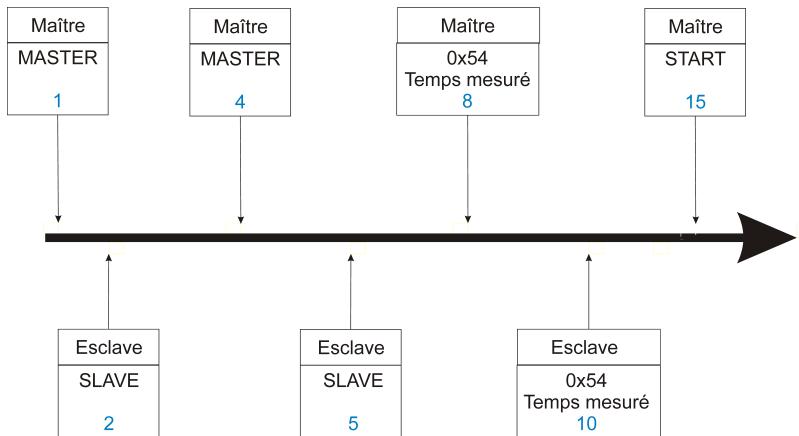


FIGURE 4.12 – Frise chronologique présentant les échanges dans le protocole GIHON entre le maître et l'esclave pour la synchronisation. Les rectangles représentent une trame GIHON avec successivement l'émetteur, la donnée et l'état de l'émetteur au moment de l'émission de la trame (chiffre coloré).

maître et 1 esclave. Une connexion RS232 permet de communiquer des informations sur les échanges d'initialisation.

Cette procédure se réalise en 138 ms par esclave.

4.6 Conclusion

Le protocole GIHON est un protocole qui respecte le protocole UDP (dans la procédure d'envoi et la procédure en cas de collisions), qui incorpore une procédure de synchronisation en cas de besoin, dont l'échange d'informations est conditionné par l'émission de PA.

Ce protocole est un puissant outil pour les expériences hybrides futures, donnant la possibilité de connecter réseaux de neurones biologique, informatique et électronique, afin de créer des plateformes neuromorphiques mettant à contribution une plus grande population de neurones, et afin d'augmenter la compatibilité des plateformes. La Table 4.2 compare le protocole GIHON et les différentes versions du protocole AER. Le protocole GIHON ne dépend pas des plateformes qu'il interconnecte.

A ce jour, seuls deux types de données sont échangés (l'identifiant du neurone ayant émis un PA et le temps auquel il est émis). Mais dans un futur proche, il est possible d'envisager de transmettre différents types d'informations. En effet, les informations transmises sont stockées dans un paquet de 32 bits (8 bits réservés + 24 bits de données). Les 8 bits réservés (voir Figure 4.7) permettent de coder jusqu'à 2^8 types d'informations différents.

Parmi les informations, nous trouvons les bouffées de PAs, les amplitudes des potentiel d'action, les potentiels de champs, etc.

		AER simple	AER à noeuds de diffusion	AER pré-structuré	AER hiarchie fractal	AER maille routage	GIHON
Adressage	Espace adressage	Direct	Direct	Local	Local	Direct	Universel
	Diffusion générale	Oui	Oui	Non	Non	Non	Oui
	Table connectivité locale définissant réseau global	Non	Non	Non	Oui	Oui	Non
	Table connectivité globale	Oui	Oui	Non	Non	Non	Non
	Neurones isolés	Oui	Non	Non	Oui	Non	Oui
Module	neurones avec synapses	Non	Oui	Oui	Non	Oui	Oui
	evenements avec poids synaptiques	Oui	Non	Non	Oui	Non	Oui
	projections avec poids synaptiques	Non	Oui	Oui	Non	Oui	Oui
	Synapses physiques	Non	Oui	Non	Oui	Non	Oui

TABLE 4.2 – Tableau inspiré et adapté de ([Zamarreno-Ramos et al., 2013](#)) comparant les différentes versions du protocole AER et notre protocole GIHON, en répertoriant le type de réseaux de neurones avec lesquels ils interagissent.

CONCLUSION & PERSPECTIVES

CETTE thèse s'est appuyée sur deux projets pluridisciplinaires : le projet européen BRAINBOW⁽¹⁾ et le projet ANR HYRENE⁽²⁾. Ces projets ont rassemblé des chercheurs appartenant au monde de la biologie, de l'électronique, des matériaux (HYRENE) et de l'informatique pour réaliser des expériences hybrides alliant l'artificiel et le vivant.

Ces trois années de recherche autour de l'hybridation se sont matérialisées au travers de deux problématiques : la construction de réseaux de neurones numériques et l'élaboration du protocole GHON. Les réseaux de neurones numériques ont été réalisés dans le cadre des deux projets pluridisciplinaires tandis que l'élaboration du protocole GHON a été faite en vue de futurs projets.

Le chapitre 1 nous a permis, dans un premier temps, de passer en revue quelques principes de base des réseaux de neurones, pour aboutir aux modèles de neurones fondés sur les mécanismes biologiques. Dans un second temps, nous avons souligné les efforts des chercheurs de l'ingénierie neuromorphe pour imiter les réseaux de neurones afin d'en étudier le fonctionnement et de créer des architectures de calculs. Nous avons ensuite présenté la double problématique de cette thèse à savoir i) la création d'un réseau de neurones en vue d'expériences hybrides et ii) la création d'un protocole de communication permettant de connecter des plateformes neuromorphiques d'origine hétérogène (Protocole GHON).

Le chapitre 2 nous a permis de présenter l'architecture de notre implémentation d'un réseau de neurones adapté au temps biologique tout en émulant plusieurs mécanismes biologiques. Parmi ces mécanismes, nous trouvons les effets AMPA-ergique et GABA-ergique (qui diminuent tous les deux l'effet d'une stimulation respectivement excitatrice ou inhibitrice dans le temps), la STP (Short Term Plasticity, mécanisme selon lequel la force d'une synapse change momentanément en fonction de l'activité du neurone pré-synaptique), le délai synaptique (reproduisant la propagation d'un PA le long de l'axone) et le bruit synaptique (généré au travers du processus d'Ornstein-Uhlenbeck). De plus, notre implémentation est configurable par un jeu d'instructions grâce à la liaison série, ce qui nous permet de décrire différents réseaux avec un même système et donc d'adapter celui-ci aux différents besoins rencontrés.

Le chapitre 3 a été divisé en trois parties. La première partie a présenté une mise en œuvre des différents mécanismes implantés. La deuxième partie montre une émulation des comportements des CPG (Central Pattern Generator) biologiques, caractérisés par des alternances de bouffées de potentiels d'action, en combinant plusieurs mécanismes implantés. Pour finir, ce chapitre s'achève sur les résultats prometteurs obtenus au sein des deux projets : l'utilisation de CPGs numériques en vue de la réhabilitation d'une moelle épinière sectionnée (projet HYRENE) et l'utilisation d'un réseau de neurones en vue de créer une jonction bidirectionnelle au-dessus d'une lésion corticale (projet BRAINBOW).

Dans le chapitre 4, nous avons passé en revue les différentes versions du protocole AER qui permettent de connecter différentes plateformes neuromorphiques. Le protocole AER avait été créé pour connecter différentes plateformes neuromorphiques afin de créer des réseaux encore plus grands, mais les différentes versions, parfois incompatibles entre elles, ont compliqué cette perspective. Cette incompatibilité nous a donné l'ambitieuse idée de concevoir un protocole de communication permettant de connecter les plateformes neuromorphiques d'origine hétérogène

¹European project FP7-ICT-2011-C

²French ANR 2010-Blan-031601

(biologique, logiciel et électronique). Ce chapitre nous a permis de montrer la faisabilité du protocole de communication GIHON, en passant par le port Ethernet.

Au travers de ces trois années de recherche, nous avons créé et implémenté une architecture biologiquement réaliste d'un réseau de neurones numérique. Cette implémentation a pu répondre aux contraintes des projets BRAINBOW et HYRENE. Par ailleurs, les résultats obtenus en connectant la plateforme MultiMed (dans laquelle notre réseau de neurones a été implanté) et une culture de neurones ou une moelle épinière furent un succès et annoncent une série d'expérimentations hybrides plus avancées (en novembre 2015 pour le projet BRAINBOW). En effet, les résultats du projet BRAINBOW vont déboucher sur l'étude des réseaux de neurones 3D. Parallèlement, les résultats du projet HYRENE vont aboutir à une collaboration avec l'entreprise *Synapse Biomedical Inc.* sur l'utilisation des CPGs *in vivo* dans l'étude de maladies neurodégénératives comme la sclérose latérale amyotrophique (SLA). Notre architecture sert aussi actuellement de support pour le dépôt de projets (projet ANR octobre 2015, KAKENHI (ANR japonaise) en novembre 2015 et un projet européen FET en avril 2016).

En plus de la poursuite de ces travaux dans le domaines des neurosciences, il sera aussi possible d'approfondir ce travail d'un point de vue électronique en cherchant à optimiser l'implémentation du code en vue de la réalisation d'une neuroprothèse faible consommation. Ce travail de thèse a permis à l'équipe AS2N d'élargir son champ de compétences dans le domaine de la conception neuromorphique en implémentant des réseaux de neurones en mode numérique. Elle envisage de poursuivre cet axe de recherche.

La communauté neuromorphique est en constante évolution avec la volonté d'agrandir la taille des réseaux de neurones. La création du protocole GIHON (**G**eneric **I**nterconnection **H**eterogeneous spiking neur**ON**s) s'inscrit dans une volonté d'élargir les possibilités d'hybridation des réseaux de neurones. GIHON ne dépend pas du type de plateformes ou de réseaux qu'il interconnecte et de ce fait permet d'interconnecter des réseaux de neurones biologiques, logiciels et/ou électroniques. Le protocole GIHON est un outil puissant qui pourra être affiné et complété avec le concours de la communauté neuromorphique.

RÉFÉRENCES BIBLIOGRAPHIQUES DE L'AUTEUR

Revue internationale avec comité de lecture

Levi T., **Ambroise M.**, Grassia F., Kohno T., Saïghi S., “Digital Spiking Neural Network for closed-loop systems”, *Seisan Kenkyu*, vol 67 :5, pp 573-579, October 2015.

Ambroise M., Levi T., Joucla S., Yvert B., Saïghi S., « Real-time biomimetic Central Pattern Generators into FPGA for hybrid experiments », *Frontiers in Neurosciences*, 7 :215, November 2013.

Chapitre d’ouvrage

Ambroise M., Levi T., Saïghi S., « Biomimetic CPG », *Biomimetic Technologies*, édité par Dr Trung Dung Ngo, *Elsevier*, 2015.

Conférences internationales avec comité de lectures et actes

Ambroise M., Levi T., Saïghi S., « Biomimetic CPG on FPGA for hybrid experiments », *International Symposium on Neuromorphic and Non-linear Engineering (ISNNE 2014)*, Tokyo, Japan, Février 2014.

Levi T., **Ambroise M.**, Grassia F., Malot O., Saïghi S., Bornat Y., Tomas J., Renaud S., “Biomimetic neural networks for hybrid experiments”, *International Symposium on Neuromorphic and Non-linear Engineering (ISNNE 2014)*, Tokyo, Japon, Février 2014.

Levi T., **Ambroise M.**, Grassia F., Saïghi S., Kohno T., Fujii T., « Biomimetics CPGs for robotic applications», *19th International Symposium on Artificial Life and Robotics*, pp. 355-358, Beppu Japon, Janvier 2014.

Bonifazi P., Massobrio P., Levi T., Difato F., Breschi G., Pasquale V., Goldin M., **Ambroise M.**, Bornat Y., Tedesco M., Bisio M., Frega M., Tessadori J., Nowak P., Grassia F., Kanner S., Ronit G., Renaud S., Matinoia S., Taverna S., Chiappalone M., « In vitro experimental and theoretical studies to restore lost neuronal functions : the Brain Bow experimental framework », *6th International IEEE EMBS Conference on Neural Engineering*, San Diego, USA, Novembre 2013.

Joucla S., **Ambroise M.**, Levi T., Lafon T., Chauvet P., Rousseau L., Lissorgues G., Saïghi S., Bornat Y., Lewis N., Renaud S., Yvert B., « Generation of Locomotor-Like Activity in the Isolated Rat Spinal Cord by Electrical Microstimulations Driven by an Artificial CPG », *6th International IEEE EMBS Conference on Neural Engineering*, San Diego, USA, Novembre 2013.

Ambroise M., Levi T., Saïghi S., « Leech Heartbeat Neural Network on FPGA », *Biomimetic and biohybrid systems, Lecture notes in computer science*, Vol. 8064, pp. 347-349, *Living Machines 2013*, London, Angleterre, 29 juillet-2 août 2013.

Ambroise M., Levi T., Bornat Y., Saïghi S., « Biorealistic Spiking Neural Network on FPGA”, *IEEE CISS 2013*, Baltimore , USA, Mars 2013.

Conférences nationales avec comité de lectures et actes

Ambroise M., Levi T., Saïghi S., « Leech Heartbeat neural network on FPGA », *Journées NeuroSTIC 2014*, Paris, 1-2 juillet 2014.

Levi T., **Ambroise M.**, Grassia F., Kohno T., “Conception d’un réseau de neurones du cœur de sangsue », *Journées Francophones de la Recherche JFR 2013*, Tokyo, Japon, Novembre 2013.

Joucla S., **Ambroise M.**, Levi T., Lafon T., Chauvet P., Rousseau L., Lissorgues G., Saïghi S., Bornat Y., Lewis N., Renaud S., Yvert B., « Generation of Locomotor-Like Activity in the isolated Rat Spinal Cord by Electrical Microstimulations Driven by an Artificial CPG », *GDR multielectrode systems & signal processing for neuroscience*, Bordeaux, France, Octobre 2013.

RÉFÉRENCES BIBLIOGRAPHIQUES

- Abbott, L. F. (1999). Lapicque's introduction of the integrate-and-fire model neuron (1907). *Brain Res Bull*, 50(5-6) :303–304.
- Abbott, L. F. and Nelson, S. B. (2000). Synaptic plasticity : taming the beast. *Nature neuroscience*, 3 Suppl(november) :1178–1183.
- Afshar, P., Khambhati, A., Stanslaski, S., Carlson, D., Jensen, R., Linde, D., Dani, S., Lazarewicz, M., Cong, P., Giftakis, J., and Others (2012). A translational platform for prototyping closed-loop neuromodulation systems. *Frontiers in neural circuits*, 6.
- Amari, S.-I. (1972). Characteristics of random nets of analog neuron-like elements. *Systems, Man and Cybernetics, IEEE Transactions on*, (5) :643–657.
- Ambroise, M., Levi, T., Bornat, Y., and Saighi, S. (2013). Biorealistic spiking neural network on FPGA. In *Information Sciences and Systems (CISS), 2013 47th Annual Conference on*, pages 1–6. IEEE.
- Arthur, J. V., Merolla, P. a., Akopyan, F., Alvarez, R., Cassidy, A., Chandra, S., Esser, S. K., Imam, N., Risk, W., Rubin, D. B. D., Manohar, R., and Modha, D. S. (2012). Building block of a programmable neuromorphic substrate : A digital neurosynaptic core. *Proceedings of the International Joint Conference on Neural Networks*, pages 10–15.
- Bamford, S. A., Hogri, R., Giovannucci, A., Taub, A. H., Herreros, I., Verschure, P., Mintz, M., and Del Giudice, P. (2012). A VLSI Field-Programmable Mixed-Signal Array to Perform Neural Signal Processing and Neural Modeling in a Prosthetic System.
- Bamford, S. A., Murray, A. F., and Willshaw, D. J. (2010). Large developing receptive fields using a distributed and locally reprogrammable address- event receiver. *IEEE Trans. Neural Netw.*, 21(2) :286–304.
- Bareket-Keren, L. and Hanein, Y. (2012). Carbon nanotube-based multi electrode arrays for neuronal interfacing : progress and prospects. *Frontiers in neural circuits*, 6.
- Barron-Zambrano, J. H. and Torres-Huitzil, C. (2013). {FPGA} implementation of a configurable neuromorphic CPG-based locomotion controller. *Neural Networks*, 45 :50–61.
- Barron-Zambrano, J. H., Torres-Huitzil, C., and Girau, B. (2010). Hardware implementation of a CPG-based locomotion control for quadruped robots. In *Artificial Neural Networks–ICANN 2010*, pages 276–285. Springer.
- Ben-Ari, Y., Khazipov, R., Leinekugel, X., Caillard, O., and Gaiarsa, J. L. (1997). GABA_A, NMDA and AMPA receptors : a developmentally regulated 'ménage à trois'. *Trends in neurosciences*, 20(11) :523–529.
- Benosman, R., Ieng, S.-H., Clercq, C., Bartolozzi, C., and Srinivasan, M. (2012). Asynchronous frameless event-based optical flow. *Neural Networks*, 27 :32–37.
- Berger, T. W., Song, D., Chan, R. H. M., Marmarelis, V. Z., LaCoss, J., Wills, J., Hampson, R. E., Deadwyler, S. A., and Granacki, J. J. (2012). A Hippocampal Cognitive Prosthesis : Multi-Input, Multi-Output Nonlinear Modeling and VLSI Implementation. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 20(2) :198–211.

- B.Girau and C.Torres-Huitzil (2007). Massively distributed digital implementation of an integrate-and-fire legion network for visual scene segmentation. *Neurocomputing*, 70(7-9) :1186–1197.
- Boahen, K. (2000). Point-to-Point connectivity between neuromorphic chips using address events. *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process*, 47(5) :416–434.
- Boahen, K. (2004). A burst-mode word-serial address-event link-I,II,III. *IEEE Trans. Circuits Syst. I, Reg. Papers*, 51(7) :1269–1280.
- Bonabi, S. Y., Asgharian, H., Bakhtiari, R., Safari, S., and Ahmadabadi, M. N. (2014). FPGA implementation of a cortical network based on the Hodgkin-Huxley neuron model. *Iconip*, 8(November) :1–12.
- Bonifazi, P., Difato, F., Massobrio, P., Breschi, G. L., Pasquale, V., Levi, T., Goldin, M., Bornat, Y., Tedesco, M., Bisio, M., and Others (2013). In vitro large-scale experimental and theoretical studies for the realization of bi-directional brain-prostheses. *Frontiers in neural circuits*, 7.
- Brainbow (2012). Brainbow Project European Union’s Seventh Framework Program (ICT-FETFP7/2007-2013, FET Young Explorers scheme) Under Grant Agreement no 284772. Disponible en ligne sur : www.brainbowproject.eu.
- Brewer, G. J., Boehler, M. D., Leondopoulos, S., Pan, L., Alagapan, S., DeMarse, T. B., and Wheeler, B. C. (2013). Toward a self-wired active reconstruction of the hippocampal trisynaptic loop : DG-CA3. *Frontiers in neural circuits*, 7.
- Bryce Beverlin, I. I. and Netoff, T. I. (2012). Dynamic control of modeled tonic-clonic seizure states with closed-loop stimulation. *Frontiers in neural circuits*, 6.
- Cassidy, A. and Andreou, A. G. (2008). Dynamical digital silicon neurons. *2008 IEEE-BIOCAS Biomedical Circuits and Systems Conference, BIOCAS 2008*, pages 289–292.
- Cassidy, A., Andreou, A. G., and Georgiou, J. (2011). Design of a one million neuron single FPGA neuromorphic system for real-time multimodal scene analysis. *2011 45th Annual Conference on Information Sciences and Systems, CISS 2011*, pages 2–7.
- Charvet, G., Billoint, O., Rousseau, L., and Yvert, B. (2007). Biomea tm : a 256-channel mea system with integrated electronics. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 171–174. IEEE.
- Chicca, E., Whatley, A. M., Lichtsteiner, P., Dante, V., Delbruck, T., Giudice, P., Douglas, R. J., and Indiveri, G. (2007). A multichip pulse-based neuromorphic infrastructure and its application to a model of orientation selectivity. *IEEE Trans. Circuits Syst. I, Reg. Papers*, 54(5) :981–993.
- Cooper, D. C. (2011). *Introduction to neuroscience I*. Donald C. Cooper Ph. D.
- Crespi, A. and Ijspeert, A. J. (2006). AmphiBot II : An amphibious snake robot that crawls and swims using a central pattern generator. In *Proceedings of the 9th international conference on climbing and walking robots (CLAWAR 2006)*, number BIOROB-CONF-2006-001, pages 19–27.

- Cymbalyuk, G. S., Gaudry, Q., Masino, M. a., and Calabrese, R. L. (2002). Bursting in leech heart interneurons : cell-autonomous and network-based mechanisms. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 22(24) :10580–10592.
- Daalen, M. V., Jeavons, P., and Shawe-Taylor, J. (1993). A stochastic neural architecture that exploits dynamically reconfigurable FPGAs. [*1993] Proceedings IEEE Workshop on FPGAs for Custom Computing Machines*.
- De Andrés, J. A., Reina, M. A., López, A., Sala-Blanch, X., and Prats, A. (2010). Blocs nerveux périphériques, paresthésies et injections intraneurales. *Praticien en Anesthésie Réanimation*, 14(4) :213–221.
- Delahaye, J.-P. (2001). Jusqu’où l’ordinateur calculera-t-il ? *Pour la Science*, pages 100–105.
- Delbrück, T. (1993). Silicon Retina with Correlation-based Velocity-Tuned Pixels. 4(3) :1–34.
- Delbrück, T. and Mead, C. A. (1991). Time-derivative adaptive silicon photoreceptor array. *International Society for Optics and Photonics*, pages 92–99.
- Delbrück, T., van Schaik, A., and Hasler, J. (2014). Research topic : neuromorphic engineering systems and applications. A snapshot of neuromorphic systems engineering. *Frontiers in Neuroscience*, 8(December) :1–2.
- Der Pol, B. (1920). A theory of the amplitude of free and forced triode vibrations. *Radio Review*, 1(1920) :701–710.
- Destexhe, a., Mainen, Z. F., and Sejnowski, T. J. (1994). An Efficient Method for Computing Synaptic Conductances Based on a Kinetic Model of Receptor Binding. *Neural Computation*, 6(1) :14–18.
- Dhingra, R. R., Zhu, Y., Jacono, F. J., Katz, D. M., Galán, R. F., and Dick, T. E. (2013). Decreased Hering–Breuer input-output entrainment in a mouse model of Rett syndrome. *Frontiers in neural circuits*, 7.
- Ekeberg, O. (1993). A combined neuronal and mechanical model of fish swimming. *Biological cybernetics*, 69(5-6) :363–374.
- Fernandez-Vargas, J., Pfaff, H. U., Rodríguez, F. B., and Varona, P. (2013). Assisted closed-loop optimization of SSVEP-BCI efficiency. *Frontiers in neural circuits*, 7.
- FitzHugh, R. (1955). Mathematical models of threshold phenomena in the nerve membrane. *The bulletin of mathematical biophysics*, 17(4) :257–278.
- FitzHugh, R. (1961). Impulses and physiological states in theoretical models of nerve membrane. *Biophysical journal*, 1(6) :445.
- Fourcaud-Trocme, N., Hansel, D., van Vreeswijk, C., and Brunel, N. (2003). How spike generation mechanisms determine the neuronal response to fluctuating inputs. *Journal of Neuroscience*, (23) :11628–11640.
- Furber, S. B., Lester, D. R., Plana, L. a., Garside, J. D., Painkras, E., Temple, S., and Brown, A. D. (2013). Overview of the SpiNNaker system architecture. *IEEE Transactions on Computers*, 62(12) :2454–2467.

- Gerstner, W. (2002). *Spiking neuron models : Single neurons, populations, plasticity.*
- Gerstner, W. and Brette, R. (2009). Adaptive exponential integrate-and-fire model. *Scholarpedia*, 4(6) :8427.
- Grassia, F., Buhry, L., Lévi, T., Tomas, J., Destexhe, A., and Saïghi, S. (2011). Tunable neuro-mimetic integrated system for emulating cortical neuron models. *Frontiers in neuroscience*, 5.
- Guggenmos, D. J., Azin, M., Barbay, S., Mahnken, J. D., Dunham, C., Mohseni, P., and Nudo, R. J. (2013). Restoration of function after brain damage using a neural prosthesis. *Proceedings of the National Academy of Sciences of the United States of America*, 110(52) :21177–82.
- Hampson, R. E., Gerhardt, G. A., Marmarelis, V., Song, D., Opris, I., Santos, L., Berger, T. W., and Deadwyler, S. (2012). Facilitation and restoration of cognitive function in primate prefrontal cortex by a neuroprosthesis that utilizes minicolumn-specific neural firing. *Journal of Neural Engineering*, 9(5) :56012.
- Hill, a. a. V., Lu, J., Masino, M. a., Olsen, O. H., and Calabrese, R. L. (2001). A model of a segmental oscillator in the leech heartbeat neuronal network. *Journal of Computational Neuroscience*, 10(3) :281–302.
- Hill, A. A. V., Masino, M. A., and Calabrese, R. L. (2002). Model of Intersegmental Coordination in the Leech Heartbeat Neuronal Network. *Journal of Neurophysiology*, pages 1586–1602.
- Hochberg, L. R., Bacher, D., Jarosiewicz, B., Masse, N. Y., Simeral, J. D., Vogel, J., Haddadin, S., Liu, J., Cash, S. S., van der Smagt, P., and Donoghue, J. P. (2012). Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. *Nature*, 485(7398) :372–375.
- Hochberg, L. R., Serruya, M. D., Friehs, G. M., Mukand, J. A., Saleh, M., Caplan, A. H., Branner, A., Chen, D., Penn, R. D., and Donoghue, J. P. (2006). Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature*, 442(7099) :164–171.
- Hodgkin, A. and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The journal of Physiology*, 117(4) :500–44.
- Hornig, C. (1984). RFC 894 : Standard for the transmission of IP datagrams over Ethernet networks. Technical report, RFC, IETF, April.
- Ijspeert, A. J. (2008). Central pattern generators for locomotion control in animals and robots : A review. *Neural Networks*, 21(4) :642–653.
- Ijspeert, A. J., Crespi, A., Ryczko, D., and Cabelguen, J.-M. (2007). From swimming to walking with a salamander robot driven by a spinal cord model. *science*, 315(5817) :1416–1420.
- Izhikevich, E. M. (2003). Simple model of spiking neurons. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, 14(6) :1569–72.
- Izhikevich, E. M. (2004). Which model to use for cortical spiking neurons ? *IEEE Transactions on Neural Networks*, 15(5) :1063–1070.
- Izhikevich, E. M. (2010). Hybrid spiking models. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 368(1930) :5061–5070.

- Izhikevich, E. M. and Edelman, G. M. (2008). Large-scale model of mammalian thalamocortical systems. *Proceedings of the national academy of sciences*, 105(9) :3593–3598.
- Jahnsen, B. Y. H. and Llinas, R. (1984). Ionic Basis for the Electroresponsiveness. pages 227–247.
- J.Lopez-Garcia, M.Moreno-Armendariz, J.Riera-Babures, M.Balsi, and X.Vilasis-Cardona (2005). RealtimevisionbyFPGAImplementedCNNs,,. *Proceedings of the 2005 European Conference on Circuit Theory and Design*, 1 :281–284.
- Joshi, S., Deiss, S., Arnold, M., Park, Y. J., and Cauwengerghs, G. (2010). Scalable event routing in hierarchical neural array architecture with global synaptic connectivity. *Proc. 12th Int. Workshop Cellular Nanoscale Networks and Their Applications*.
- Joucla, S., Ambroise, M., Levi, T., Lafon, T., Chauvet, P., Rousseau, L., Lissorgues, G., Saïghi, S., Bornat, Y., Lewis, N., and Others (2013). Generation of Locomotor-Like Activity in the Isolated Rat Spinal Cord by Electrical Microstimulations Driven by an Artificial CPG. In *GDR multielectrode systems & signal processing for neuroscience*.
- Jung, R. (2011). *Biohybrid Systems*.
- Jung, R., Brauer, E. J., and Abbas, J. J. (2001). Real-time Interaction Between a Neuromorphic Electronic Circuit and the Spinal Cord. 9(3) :0–2.
- Jung, R., Horch, K., Abbas, J. J., Phillips, S., Bakkaloglu, B., and Kim, S.-J. (2015). Communication interface for sensory stimulation.
- Kandel, E., Schwartz, J. H., and Jessell, T. M. (2000). *Principles of Neural Science*. Medical, Mc Graw-Hill, 4th edition.
- Katz, B. and Miledi, R. (1965). The effect of calcium on acetylcholine release from motor nerve terminals. *Proceedings of the Royal Society of London B : Biological Sciences*, 161(985) :496–503.
- Khan, M., Lester, D., Plana, L., Rast, A., Jin, X., Painkras, E., and Furber, S. (2008). SpiNNaker : Mapping neural networks onto a massively-parallel chip multiprocessor. *Proceedings of the IEEE Int. Joint Conf. Neural Networks*, pages 2849–2856.
- Kolbl, F. (2014). *Design of electrical adaptive stimulators for different pathological contexts : a global approach*. PhD thesis.
- Krips, M., Lammert, T., and Kummert, A. (2002). FPGA implementation of a neural network for a real-time handtracking system. *Proceedings of First IEEE International Workshop on Electronic Design Test and Applications*, pages 313–317.
- Kuga, N., Sasaki, T., Takahara, Y., Matsuki, N., and Ikegaya, Y. (2011). Large-scale calcium waves traveling through astrocytic networks in vivo. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 31(7) :2607–2614.
- Le Masson, G., Renaud-Le Masson, S., Debay, D., and Bal, T. (2002). Feedback inhibition controls spike transfer in hybrid thalamic circuits. *Nature*, 417(6891) :854–858.

- Lee, Y. J., Lee, J., Kim, K. K., Kim, Y.-B., and Ayers, J. (2007). Low power CMOS electronic central pattern generator design for a biomimetic underwater robot. *Neurocomputing*, 71(1-3) :284–296.
- Lewis, M., Hartmann, M. J., Etienne-Cummings, R., and Cohen, A. H. (2001). Control of a robot leg with an adaptive aVLSI CPG chip. *Neurocomputing*, 38–40 :1409–1421.
- Li, C. A. I., Lowe, R., and Ziemke, T. (2013). Humanoids Learning to Walk : a Natural CPG-Actor-Critic Architecture. *Frontiers in Neurorobotics*, 7(5).
- Li, W.-C., Merrison-Hort, R., Zhang, H.-Y., and Borisuk, R. (2014). The generation of antiphase oscillations and synchrony by a rebound-based vertebrate central pattern generator. *The Journal of Neuroscience*, 34(17) :6065–6077.
- Lin, J., Merolla, P., Arthur, J., and Boahen, K. (2006). Programmable connections in neuromorphic grids. *Proc. Int. Midwest Symp. Circuits and Systems*, pages 80–84.
- Linares-Barranco, B., Sánchez-Sinencio, E., Rodríguez-Vázquez, A., and Huertas, J. L. (1993). A CMOS analog adaptive BAM with on-chip learning and weight refreshing. *Neural Networks, IEEE Transactions on*, 4(3) :445–455.
- Liu, C.-N., Michaelis, M., Amir, R., and Devor, M. (2000). Spinal nerve injury enhances subthreshold membrane potential oscillations in DRG neurons : relation to neuropathic pain. *Journal of neurophysiology*, 84(1) :205–215.
- Liu, S. C., Kramer, J., Indiveri, G., Delbrück, T., Burg, T., and Douglas, R. (2001). Orientation-selective aVLSI spiking neurons. *Neural Networks*, 14(6-7) :629–643.
- Mahowald, M. and Douglas, R. (1991a). A silicon neuron. *Nature*, 354(6354) :515–518.
- Mahowald, M. and Douglas, R. (1991b). A silicon neuron. *Nature*, 354(6354) :515–518.
- Matsuoka, K. (1987). Mechanisms of frequency and pattern control in the neural rhythm generators. *Biological Cybernetics*, 56(5-6) :345–353.
- Mayr, C., Eisenreich, H., Henker, S., and Schüffny, R. (2005). Pulsed multilayered image filtering : A VLSI implementation. *Int. J. Appl. Math. Comput. Sci.*, 1 :60–65.
- Mead, C. (1989). Analog VLSI and Neural. *Addison-Wesley Longman Publishing Co., Inc.*
- Merolla, P., Arthur, J., Shi, B., and K. Boahen (2007). Expandable networks for neuromorphic chips. *IEEE Trans. Circuits Syst. I, Reg. Papers.*, 54(2) :301–311.
- Minneci, F., Kanichay, R. T., and Silver, R. A. (2012). Estimation of the time course of neurotransmitter release at central synapses from the first latency of postsynaptic currents. *Journal of neuroscience methods*, 205(1) :49–64.
- Misra, J. and Saha, I. (2010). Artificial neural networks in hardware : A survey of two decades of progress. *Neurocomputing*, 74(1-3) :239–255.
- Mortara, A., Vittoz, E. A., and Venier, P. (1995). A communication scheme for Systems, analog VLSI perceptive. *IEEE J. Solid-State Circuits*, 30(6) :660–669.

- Nadim, F., Olsen, O. H., De Schutter, E., and Calabrese, R. L. (1995). Modeling the leech heartbeat elemental oscillator I. Interactions of intrinsic and synaptic currents. *Journal of computational neuroscience*, 2(3) :215–235.
- Nagumo, J., Arimoto, S., and Yoshizawa, S. (1962). An active pulse transmission line simulating nerve axon. *Proceedings of the IRE*, 50(10) :2061–2070.
- Nakada, K., Asai, T., and Amemiya, Y. (2003). An analog CMOS central pattern generator for interlimb coordination in quadruped locomotion.
- Nazari, S., Amiri, M., Faez, K., and Amiri, M. (2015a). Multiplier-less digital implementation of neuron–astrocyte signalling on FPGA. *Neurocomputing*, 164 :281–292.
- Nazari, S., Faez, K., Amiri, M., and Karami, E. (2015b). A digital implementation of neuron–astrocyte interaction for neuromorphic applications. *Neural Networks*, 66 :79–90.
- Nicolelis, M. A. L. and Lebedev, M. A. (2009). Principles of neural ensemble physiology underlying the operation of brain-machine interfaces. *Nat Rev Neurosci*, 10(7) :530–540.
- Nishimura, Y., Perlmutter, S. I., and Fetz, E. E. (2013). Restoration of upper limb movement via artificial corticospinal and musculospinal connections in a monkey with spinal cord injury. *Frontiers in neural circuits*, 7.
- Olsen, O. H. and Calabrese, R. L. (1996). Activation of intrinsic and synaptic currents in leech heart interneurons by realistic waveforms. *The Journal of Neuroscience*, 16(16) :4958–4970.
- Olsen, O. H., Nadim, F., and Calabrese, R. L. (1995). Modeling the leech heartbeat elemental oscillator II. Exploring the parameter space. *Journal of computational neuroscience*, 2(3) :237–257.
- Opris, I., Fuqua, J. L., Huettl, P. F., Gerhardt, G. A., Berger, T. W., Hampson, R. E., and Deadwyler, S. A. (2012). Closing the loop in primate prefrontal cortex : inter-laminar processing. *Frontiers in neural circuits*, 6.
- Papa, M., C., D. L., Petta, F., Alberghina, L., and Cirillo, G. (2014). Astrocyte-neuron interplay in maladaptive plasticity.
- Pimashkin, A., Gladkov, A., Mukhina, I., and Kazantsev, V. (2013). Adaptive enhancement of learning protocol in hippocampal cultured networks grown on multielectrode arrays. *Frontiers in neural circuits*, 7.
- Pocek, K., Tessier, R., and DeHon, A. (2013). Birth and Adolescence of Reconfigurable Computing : A Survey of the First 20 Years of Field-Programmable Custom Computing Machines. *Ieee*, page 17.
- Pospischil, M., Toledo-Rodriguez, M., Monier, C., Piwkowska, Z., Bal, T., Frégnac, Y., Markram, H., and Destexhe, A. (2008). Minimal hodgkin-huxley type models for different classes of cortical and thalamic neurons. *Biological Cybernetics*, 99 :427–441.
- Potter, S. M., El Hady, A., and Fetz, E. E. (2014). Closed-loop neuroscience and neuroengineering. *Frontiers in Neural Circuits*, 8(September) :2013–2015.

- Protocol, U. D. (1980). RFC 768 J. Postel ISI 28 August 1980. *Isi.*
- Prueckl, R., Taub, A. H., Herreros, I., Hogri, R., Magal, A., Bamford, S. A., Giovannucci, A., Almog, R. O., Shacham-Diamand, Y., Verschure, P., Mintz, M., Scharinger, J., Silmon, A., and Guger, C. (2011). Behavioral rehabilitation of the eye closure reflex in senescent rats using a real-time biosignal acquisition system. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pages 4211–4214.
- Quotb, A. (2012). *Systems and methods for adaptive and real-time detection of biological activity*. Theses, Universit{é} Sciences et Technologies - Bordeaux I.
- Rast, A. D., Partzsch, J., Mayr, C., Schemmel, J., Hartmann, S., Plana, L. a., Temple, S., Lester, D. R., Schuffny, R., and Furber, S. (2013). A location-independent direct link neuromorphic interface. *Proceedings of the International Joint Conference on Neural Networks*.
- Reuveni, I., Fried Man, A., Amitai, Y., and Gutnick, M. (1993). Stepwise repolarization from ca₂₊ plateaus in neocortical pyramidal cells : evidence for nonhomogeneous distribution of hva ca₂₊ channels in dendrites. *The Journal of Neuroscience*, 13(11) :4609–21.
- Rice, K., Bhuiyan, M., Taha, T., Vutsinas, C., and Smith, M. (2009). FPGA Implementation of Izhikevich Spiking Neural Networks for Character Recognition. *2009 International Conference on Reconfigurable Computing and FPGAs*, pages 451–456.
- Robinson, J. T., Jorgolli, M., and Park, H. (2013). Nanowire electrodes for high-density stimulation and measurement of neural circuits. *Frontiers in neural circuits*, 7.
- Ros, E., Ortigosa, E. M., Agís, R., Carrillo, R., and Arnold, M. (2006). Real-time computing platform for spiking neurons (RT-spike). *IEEE Transactions on Neural Networks*, 17(4) :1050–1063.
- Rudolph, M. and Destexhe, A. (2005). An extended analytic expression for the membrane potential distribution of conductance-based synaptic noise. *Neural Computation*, 17(11) :2301–2315.
- Sabarad, J., Kestur, S., Park, M. S., Dantara, D., Narayanan, V., Chen, Y., and Khosla, D. (2012). A reconfigurable accelerator for neuromorphic object recognition. *Proceedings of the Asia and South Pacific Design Automation Conference, ASP-DAC*, pages 813–818.
- Schmidhuber, J. (2015). Deep learning in neural networks : An overview. *Neural Networks*, 61 :85–117.
- Schoenauer, T., Atasoy, S., Mehrtash, N., and Klar, H. (2002). NeuroPipe-Chip : A digital neuro-processor for spiking neural networks. *IEEE Transactions on Neural Networks*, 13(1) :205–213.
- Serrano-gotarredona, R., Oster, M., Lichtsteiner, P., Linares-barranco, A., Paz-vicente, R., Gómez-rodríguez, F., Camuñas mesa, L., Berner, R., Rivas-pérez, M., Delbrück, T., Liu, S.-c., Douglas, R., Häfliger, P., Jiménez-moreno, G., Ballcells, A. C., Serrano-gotarredona, T., Acosta-jiménez, A. J., and Linares-barranco, B. (2009). CAVIAR : A 45k Neuron , 5M Synapse , 12G Connects / s AER Hardware Sensory – Processing – Learning – Actuating System for High-Speed Visual Object Recognition and Tracking. 20(9) :1417–1438.

- Sorensen, M., DeWeerth, S., Cymbalyuk, G., and Calabrese, R. L. (2004). Using a hybrid neural system to reveal regulation of neuronal network activity by an intrinsic current. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 24(23) :5427–5438.
- Stein, R. B. (1967). Some models of neuronal variability. *Biophysical journal*, 7(1) :37.
- Stevenson, P. A. and Kutsch, W. (1987). A reconsideration of the central pattern generator concept for locust flight. *Journal of Comparative Physiology A*, 161(1) :115–129.
- Still, S. and Tilden, M. W. (1998). Controller for a Four-Legged Walking Machine. In *Neuromorphic Systems*, volume Volume 10 of *Progress in Neural Processing*, pages 138–148. WORLD SCIENTIFIC.
- Tessadori, J., Bisio, M., Martinoia, S., and Chiappalone, M. (2012). Modular neuronal assemblies embodied in a closed-loop environment : toward future integration of brains and machines. *Frontiers in neural circuits*, 6.
- Torres-Huitzil, C. and Girau, B. (2008). Implementation of central pattern generator in an FPGA-based embedded system. In *Artificial Neural Networks-ICANN 2008*, pages 179–187. Springer.
- Touboul, J. (2004). Bifurcation analysis of a general class of non-linear integrate and fire neurons. pages 1–36.
- Touboul, J. (2008). Bifurcation analysis of a general class of nonlinear integrate-and-fire neurons. 68(4) :1045–1079.
- Tritsch, D., Chesnoy-Marchais, D., and Feltz, A. (1999). *Physiologie du neurone*. France. Wolters Kluwer France.
- Van der Pol, B. and Van der Mark, J. (1928). LXXII. The heartbeat considered as a relaxation oscillation, and an electrical model of the heart. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 6(38) :763–775.
- Vogelstein, R., Tenore, F., Etienne-Cummings, R., Lewis, M., and Cohen, A. (2006). Dynamic control of the central pattern generator for locomotion. *Biological Cybernetics*, 95(6) :555–566.
- Vogelstein, R. J., Mallik, U., Vogelstein, J. T., and Cauwenberghs, G. (2007). Dynamically reconfigurable silicon array of spiking neurons with conductance-based synapses. *IEEE Transactions on Neural Networks*, 18(1) :253–265.
- Walter, A., Murguialday, A. R., Rosenstiel, W., Birbaumer, N., and Bogdan, M. (2012). Coupling BCI and cortical stimulation for brain-state-dependent stimulation : methods for spectral estimation in the presence of stimulation after-effects. *Frontiers in neural circuits*, 6.
- Walter, A., Murguialday, A. R., Spüler, M., Naros, G., Leão, M. T., Gharabaghi, A., Rosenstiel, W., Birbaumer, N., and Bogdan, M. (2013). Erratum : Coupling BCI and cortical stimulation for brain-state-dependent stimulation : methods for spectral estimation in the presence of stimulation after-effects. *Frontiers in neural circuits*, 7.
- Wang, D. and Terman, D. (1997). Image segmentation based on oscillatory correlation. *Neural computation*, 9(4) :805–836.

- Wang, R., Cohen, G., Stiefel, K. M., Hamilton, T. J., Tapson, J., and van Schaik, A. (2013). An FPGA Implementation of a Polychronous Spiking Neural Network with Delay Adaptation. *Frontiers in Neuroscience*, 7(February) :1–14.
- Weeks, M., Freeman, M., Moulds, A., and Austin, J. (2005). Developing Hardware-Based Applications Using Presence-2. pages 107–114.
- Wijekoon, J. H. B. and Dudek, P. (2008). Compact silicon neuron circuit with spiking and bursting behaviour. *Neural Networks*, 21(2) :524–534.
- Yamada, W. M., Koch, C., and Adams, P. R. (1989). *Methods in neuronal modeling*. Cambridge, MA, USA.
- Yang, F. and Paindavoine, M. (2003). Implementation of an RBF neural network on embedded systems : real-time face tracking and identity verification. *IEEE Transaction on Neural Networks*, 14(5) :1162–1175.
- Zamarreno-Ramos, C., Linares-Barranco, A., Serrano-Gotarredona, T., and Linares-Barranco, B. (2013). Multicasting mesh AER : A scalable assembly approach for reconfigurable neuromorphic structured AER systems. Application to ConvNets. *IEEE Transactions on Biomedical Circuits and Systems*, 7(1) :82–102.
- Zielinska, T. (1996). Coupled oscillators utilised as gait rhythm generators of a two-legged walking machine. *Biological Cybernetics*, 74(3) :263–273.
- Zou, Q., Bornat, Y., Tomas, J., Renaud, S., and Destexhe, A. (2006). Real-time simulations of networks of Hodgkin-Huxley neurons using analog circuits. *Neurocomputing*, 69(10-12) :1137–1140.