HAL
open science

Conception d'une plateforme modulable de réseau de neurones biomimétiques pour l'étude des maladies neurodégénératives

Farad Khoyratee

To cite this version:

Farad Khoyratee. Conception d'une plateforme modulable de réseau de neurones biomimétiques pour l'étude des maladies neurodégénératives. Sciences de l'ingénieur [physics]. Université de Bordeaux, 2019. Français. NNT : . tel-02527385

HAL Id: tel-02527385

https://hal.science/tel-02527385

Submitted on 1 Apr 2020

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

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

Thèse

PRÉSENTÉE À

L'Université de Bordeaux

POUR OBTENIR LE GRADE DE

DOCTEUR DE L'UNIVERSITÉ DE BORDEAUX

ÉCOLE DOCTORALE DES SCIENCES PHYSIQUES ET DE L'INGÉNIEUR

SPÉCIALITÉ : Électronique

par Farad KHOYRATEE

Conception d'une plateforme modulable de réseau de neurones biomimétiques pour l'étude des maladies neurodégénératives

Sous la direction de Sylvain Saïghi et Timothée Levi

Soutenue le : 13 Décembre 2019

Après avis de :

Benoît Miramond	Professeur des Universités, Université de Côte d'Azur	Rapporteur
Mustapha Nadi	Professeur des Universités, université de Lorraine	Rapporteur

Devant la commission d'examen formée de:

Benoît Miramond	Professeur des Universités, Université de Côte d'Azur	Président
Mustapha Nadi	Professeur des Universités, Université de Lorraine	Rapporteur
Florian Kolbl	Maître de conférences, Université de Cergy	Membre
Dominique Dallet	Professeur des Universités, Université de Bordeaux	Membre
Takashi Kohno	Professeur des Universités, Université de Tokyo	Membre invité
Timothée Leleu	Project associate professor , Université de Tokyo	Membre invité
Timothée Levi	Maître de conférences, Université de Bordeaux	Co-directeur de thèse
Sylvain Saïghi	Maître de conférences, Université de Bordeaux	Directeur de thèse

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

Mes travaux de thèses ont été le fruit d'un long travail qui n'a seulement été possible que grâce à certaines personnes sans lesquelles le déroulement de la thèse n'aurait très probablement pas été le même. Ils ont participé à me soutenir, me faire évoluer, me donner la passion de la recherche, m'inspirer et ils m'ont conseillé sur mes choix de vie et de carrière.

Tout d'abord, Je souhaite remercier Timothée Levi qui a co-encadré ma thèse. Il a tout le temps été présent, m'a donné de nombreux et précieux conseils, m'a soutenu et inspiré. Il mène ses travaux de recherche avec passion tout en y ajoutant une dimension humaine que je ne peux qu'apprécier et respecter, ce qui, a rendu la thèse intéressante et agréable.

Je remercie également Sylvain Saïghi pour ses conseils avisés tout le long de la thèse en sa qualité de directeur de thèse, et qui, a toujours su répondre présent à tout instant. Son honnêteté ainsi que sa rigueur m'ont poussé à aller toujours plus loin.

Je souhaiterais ensuite remercier Benoît Miramond, président de Jury et Rapporteur, pour avoir pris le temps de lire ce manuscrit, ses commentaires et la considération qu'il a portée à mon travail. Je remercie également, Mustapha Nadi du temps passé sur la lecture et les commentaires du manuscrit. Je remercie Florian Kolbl d'avoir pris le temps d'assister à ma soutenance de thèse en tant que membre du jury. Enfin, je remercie Dominique Dallet pour nos interactions positives, l'aide qu'il m'a apporté et sa participation à la soutenance en tant que membre de jury.

L'équipe AS2N possède une ambiance particulière générée par Jean Tomas que je remercie pour son humour et l'aura de détente qu'il instaure. Il a aussi été une grande source de connaissance, tous domaines confondus, qu'il n'a pas hésité à partager. Je remercie aussi Adrien Vincent pour ses précieux conseils et aides apportés. J'aimerais remercier les doctorants de l'équipe dont Charly Meyer avec qui un lien d'amitié a pu se créer. Je le remercie de m'avoir soutenu. Je remercie également Pierre Lewden, qui a participé à l'ambiance de l'équipe par sa personnalité et ses compétences pendant de nombreuses années. Nos discussions du midi ont su être animées, très animées grâce à ces personnes.

L'équipe AS2N se situe dans le groupe bioélectronique, groupe dont je remercie toutes les personnes qui la composent pour avoir instauré une atmosphère détendue et emplie de bonne humeur. Parmi ces personnes, Je souhaiterais remercier Yannick Bornat qui n'a pas hésité à mettre son travail en pause afin de répondre à mes questions techniques avec détail, patience et pédagogie.

Je remercie les personnes du laboratoire IMS et particulièrement Fabienne Prévot du service RH pour son amitié, nos longues conversations, et ses conseils administratifs. Je remercie Jean-Marie Pham pour son accueil et ses conseils au sein de l'IUT Mesures Physiques. J'aimerais également remercier Yann Deval de m'avoir accueilli au sein du laboratoire IMS.

J'ai également reçu un accueil chaleureux au sein de l'université de Tokyo grâce à plusieurs personnes. Parmi elles, Je souhaite remercier Takashi Kohno d'avoir participé à ma soutenance de thèse en tant que membre de Jury et d'avoir mis son laboratoire, au Japon, à ma libre disposition. Je remercie vivement Timothée Leleu avec qui j'ai collaboré lors de ma thèse et avec qui je continue de collaborer. Il m'a beaucoup appris et m'a offert de grandes opportunités de carrière. Je tiens à remercier Kazuyuki Aihara qui, malgré ses grandes res-

ponsabilités, m'a accordé du temps, financé mes voyages et accueilli avec une grande gentillesse. Finalement, je tiens à remercier Agnès Tixier-Mita avec qui j'ai eu plaisir à collaborer. Elle m'a toujours accueilli avec le sourire et a toujours mis à disposition tout le matériel de son laboratoire.

Je remercie chaleureusement mes prédécesseurs Filippo Grassia et Matthieu Ambroise pour leurs disponibilités et leurs partages. Leurs travaux ont servi de bases aux travaux de thèse présentés dans ce manuscrit.

J'ai aussi reçu un grand soutien de stagiaires avec lesquels je me suis lié d'amitié. Je remercie Clément Chapuzet qui a participé à la partie électronique du projet, Louis Chatenet, qui s'est occupé de la partie simulation des synapses et Adrien Bernard, de la partie informatique. Ils ont été présents pendant ma thèse et à ma soutenance de thèse et leurs travaux servent toujours et serviront aux prochaines générations. Je remercie aussi Carole-Anne Volette avec qui on a pu collaborer sur la partie biologie mais aussi sur la découverte des restaurants, bars et autres originalités de Tokyo.

Bien entendu, je remercie grandement mes parents et mes sœurs qui ont toujours cru en moi et m'ont toujours soutenu en toutes circonstances.

Un de mes plus grands supports depuis le collège, un soutien moral inégalé, des conseils inestimables et surtout, beaucoup de rires. Un grand merci à Cédric Marseille, ami de longue date et complice de tous méfaits.

Enfin, je souhaite remercier Kazune Okubo, une des personnes qui m'a le plus soutenu au quotidien au Japon ou en France avec le sourire et la bonne humeur.

Résumé

Conception d'une plateforme modulable de réseau de neurones biomimétiques pour l'étude des maladies neurodégénératives

Résumé :

Les neurosciences ont fait l'objet de nombreuses études et ont vu émerger de nouveaux domaines de recherche où la technologie et la biologie peuvent être utilisées en synergie dans le but de trouver des solutions pour comprendre et guérir les maladies neurologiques. Ces maladies affectent des millions de personnes à travers le monde. L'Organisation Mondiale de la Santé (OMS) prévoit une multiplication par 3 du nombre de malades dans les 30 ans à venir.

Les progrès des neurosciences ont permis l'émergence de modèles décrivant la physiologie des neurones et aussi de méthodes d'implémentation matérielle de ces modèles. Parmi ces méthodes, les neuroprothèses sont des dispositifs permettant de rétablir certaines fonctions neuronales grâce à une communication avec le système nerveux. La méthodologie de conception implique l'étude du comportement des cellules et la création de plateformes biomimétiques interagissant en temps réel avec les cellules vivantes.

Dans ces travaux de thèse, la réalisation du système biomimétique a été menée grâce à des composants numériques tels que les Field Programmable Gate Arrays (FPGA), ce qui permet de bénéficier de la flexibilité et de la rapidité de prototypage de ces technologies. La plateforme temps réel de réseaux de neurones biologiquement réalistes développée est paramétrable. Elle devient ainsi un outil neuro-computationnel permettant la réalisation d'expériences bio-hybrides pour l'étude du comportement du système nerveux et plus particulièrement des maladies neurodégénératives.

Ces travaux se sont placés dans un contexte plus large. La bibliothèque d'opérateurs numériques sur FPGA développée pour la plateforme a été réutilisée pour l'étude des dynamiques semblables à celles des réseaux de neurones telles que la simulation de réseaux biochimiques ou la résolution de problèmes d'optimisation combinatoire.

Mots-clés : FPGA, Réseau de neurones, Bio-hybrides, Maladies neurodégénératives

Abstract

Design of a modular biomimetic neural network for the study of neurodegenerative diseases

Abstract :

Neuroscience has been the subject of many studies and has seen new fields of research emerge where technology and biology can be used to find solutions to understand and cure neurological diseases. These illnesses affect millions of people around the world. The World Health Organization (WHO) predicts a 3 fold increase in the number of patients in the next 30 years.

Advances in neuroscience have led to the development of models describing the physiology of neurons and also methods of hardware implementation of these models. Among these methods, neuroprostheses are devices for restoring certain neuronal functions through communication with the nervous system.

This thesis work show that the realization of the biomimetic system was carried out thanks to digital components such as Field Programmable Gate Array (FPGA) which allows to benefit from the flexibility and speed of prototyping of these technologies. The real-time platform of biologically realistic neural networks developed is configurable. It becomes a neuro-computational tool allowing the realization of bio-hybrid experiments for the study of the behavior of the nervous system and more particularly of the neurodegenerative diseases.

This work was placed in a larger context. The FPGA digital operator library developed for the platform has been reused for the study of dynamics similar to neural networks such as biochemical network simulation or combinatorial optimization problem solving.

Keywords : FPGA, Neural network, Bio-hybrid, Neurological diseases

Table des matières

Introduction **4**

1 Du cerveau à la machine **8**
1.1 Introduction 9
1.2 Morphologie du système nerveux 9
1.2.1 Neurone 9
1.2.2 Dendrites 11
1.2.3 Axones 11
1.2.4 Synapses 11
1.2.5 Membrane plasmique 11
1.3 Transmission de l'information 12
1.3.1 Formation des potentiels d'action 12
1.3.2 Transmission des influx nerveux 13
1.3.3 Variabilité 18
1.3.4 Le codage des informations 19
1.4 Maladies neurologiques 21
1.4.1 Un aperçu des maladies neurodégénératives 21
1.4.2 Étude des maladies neurologiques 22
1.5 Modèle de neurones 24
1.5.1 Hodgkin-Huxley : Un modèle bioréaliste ou biodétaillé ? 25
1.5.2 Quel modèle choisir ? 26
1.6 Modèles Biomimétiques 28
1.6.1 Classification des neurones 28
1.6.2 Les limites du modèle de HH et l'importance des compartiments 30
1.6.3 Les niveaux de détail 32
1.7 La modélisation biodétaillée 32
1.7.1 Choix du modèle 32
1.7.2 Modèle à compartiments multiples 32
1.7.3 Modèle de synapses 34
1.7.4 Modèles de bruit 35
1.8 Résumé 35

2 Implémentation matérielle de réseaux de neurones **36**
2.1 Introduction 37
2.2 Contexte Technologique 37
2.2.1 État de l'art des implémentations de modèles bioréalistes 37
2.2.2 Les implémentations FPGA 38
2.2.3 Méthode de simplification 39
2.2.4 Le cas du FPGA choisi 40

2.2.5 Paramètres de [POSP 2008] 40
2.2.6 Adapter les mathématiques au numérique 42
2.3 Réseau de neurones 44
2.3.1 Synapses 44
2.3.2 Bruit 46
2.4 Validation 47
2.4.1 Comparer des potentiels d'action 47
2.4.2 Quantifier la différence 49
2.4.3 Analyse de systèmes dynamiques 51
2.4.4 Stabilité/instabilité 52
2.4.5 « Limit cycle » et analyse de bifurcations 52
2.5 Modèles à compartiments 53
2.5.1 Èquations du modèle 53
2.5.2 Description des courants ioniques 54
2.5.3 Effet des compartiments 56
2.5.4 Implémentation 56
2.6 Résumé 59

3 De la bibliothèque VHDL aux systèmes pour l'ingénierie neuromorphique 61
3.1 Introduction 62
3.2 Codage et précision 62
3.3 Temps de propagation et pipeline 63
3.4 Multiplication et Division 63
3.4.1 Multiplication 63
3.4.2 Division 65
3.5 Fonctions hyperboliques et exponentielles 66
3.5.1 CORDIC 67
3.5.2 Interpolation de Lagrange 68
3.5.3 Comparatif 70
3.6 Résolution d'équations différentielles 72
3.7 Produit matrice vecteur 73
3.7.1 État de l'art 73
3.7.2 Méthodes proposées 74
3.8 Protocoles de communication 80
3.8.1 UART 80
3.8.2 Visualiser les potentiels d'action avec un CNA 80
3.9 Architecture des réseaux de neurones 81
3.9.1 Architecture générale 81
3.9.2 Architecture spécifique 82
3.9.3 Ressources 84
3.10 Résumé 85

4 Application de la plateforme de calcul neuromorphique 86
4.1 Introduction 87
4.2 *Spiking Neural Network* : Validation 87
4.2.1 Neurones corticaux 87
4.2.2 Bruit 88
4.2.3 Synapses 89
4.2.4 Comportement de réseaux 91

4.2.5 Simulation de maladies 93
4.2.6 Interface graphique 94
4.3 Expériences Biohybrides 95
4.3.1 Les organoïdes, une solution pour étudier le cerveau ? 95
4.3.2 La microfluidique et la biocompatibilité des matériaux 99
4.3.3 Système expérimental pour l'étude de cellules biologiques 100
4.4 Application de la bibliothèque VHDL pour ingénierie neuromorphique 101
4.4.1 Modèle proie-prédateur 102
4.4.2 Problème d'optimisation combinatoire 103
4.5 Résumé 104

Conclusion **105**

Bibliographie **109**

Publications de l'auteur **125**

Annexes **129**
Field Programmable Gate Array (FPGA) 130
Temps de propagation et pipeline 134
Modèle proie-prédateur 138
Optimisation combinatoire 140

Introduction

Neurosciences, composition des termes *neuro*, qui a pour sujet les neurones ou le système nerveux, et *sciences*, qui a trait à la connaissance. Neurosciences se décline en une multitude de termes reflétant un accès de la discipline à des spécialités diverses et variées. Neurophysiologie, neuroanatomie, neurologie, neurosciences cognitives, neurosciences computationnelles, tant de disciplines montrant la richesse et la complexité du sujet. Les neurosciences ont fait naître dans l'imaginaire collectif une riche culture littéraire et cinématographique, et ont implanté mythes et légendes mais ont aussi donné de nombreux sujets de questionnement dans la communauté scientifique. Les neurosciences computationnelles sont une association de termes démontrant un désir de calculer le système nerveux, de programmer, voire de simuler le cerveau. Une question se pose : à quel niveau de détail devons-nous simuler le vivant ? Les interactions biochimiques des différentes cellules, la transmission des informations du nerf optique, la genèse d'un neurone, sa mort ou certaines maladies neurodégénératives ?

Les maladies neurologiques sont un fléau qui touche, selon l'OMS, des centaines de millions de personnes à travers le monde et qui ne cesse d'augmenter. ǎ l'heure où les chercheurs espèrent trouver des procédés pharmaceutiques comme traitement des maladies ou des méthodes d'imageries pour les diagnostics, des alternatives biologiques, matérielles ou technologiques sont proposées comme la simulation des comportements biologiques sur systèmes électroniques ou numériques. Cependant, les limitations technologiques actuelles ne permettent pas de tout simuler et la limite de nos connaissances de définir ce qu'est le tout. La simulation du cerveau, ou plutôt de certaines parties du cerveau, amène aussi au postulat suivant : s'il est possible de reproduire les activités du cerveau, de comprendre le cerveau, peut-être est-il possible de communiquer directement avec lui pour en étudier les mécanismes.

Une manière d'étudier les mécanismes du cerveau se fait par l'étude de la transmission de l'information. Le système nerveux communique par transmission d'influx nerveux et peut être reproduit par simulation. La modélisation a permis l'émergence de technologies d'enregistrement des activités du système nerveux [POTT 2014], de théories et la découverte de phénomènes [RECC 1999]. Comprendre comment le cerveau communique a aussi permis des interactions avec celui-ci [JOUC 2016] dans le cadre de prothèses mécaniques [VELL 2008], d'interaction biohybride entre un réseau de neurones artificiels et un réseau de neurone biologiques pour l'étude du comportement des nerfs rétiniens [LEMA 2002], de réparations des lésions de la moelle épinière [JACK 2016] ou de maladies neurologiques [HAMM 2007]. Ces technologies ont pour base l'électronique, ce qui pose un certain problème lors de la connexion vivant/artificiel au vu du caractère non biocompatible des conducteurs. Pour pallier ce problème, d'autres disciplines s'attellent à reproduire les interactions chimiques des neurones à travers les systèmes microfluidiques [KHOYb 2019].

Ce manuscrit de thèse décrit notamment les simulations ou plus précisément la reproduction des activités neuronales, car il s'agit d'une spécialité de l'équipe Architecture on Silicon of Neural Network (AS2N) du laboratoire Intégration du Matériau vers le Système (IMS) de l'Université de Bordeaux au sein duquel les travaux se sont déroulés.

Les sciences physiques se sont construites sur la base de modèles décrivant des phénomènes de la nature. Le langage universel compréhensible par les scientifiques du domaine quel que soit l'endroit dans le monde est basé sur les mathématiques. William Thomson,

alias Lord Kelvin, disait à ce sujet : “I am never content until I have constructed a mathematical model of what I am studying. If I succeed in making one, I understand; otherwise I do not” [THOM 1904]. Lorsque l’on parle de modèles de neurones ou de réseaux de neurones, l’intelligence artificielle (IA) est ce qui revient le plus souvent. L’IA utilise des modèles bio inspirés pour le calcul (reconnaissance d’images, traitement de données, etc.). Il existe cependant un autre type englobant les modèles dits biomimétiques, c’est-à-dire, qui reproduisent la biologie.

Le modèle le plus complet décrivant le mécanisme des neurones et reproduisant les potentiels d’action remonte aux années cinquante avec les travaux d’Alan Lloyd Hodgkin et Andrew Fielding Huxley sur l’axone géant de calmar. Ce modèle reste encore aujourd’hui le plus complet mais aussi le plus complexe. Ainsi, plusieurs simplifications et autres modèles plus simples ont vu le jour. Parmi eux on peut citer les modèles Integrate and Fire ou Izhikevich. Cependant, le modèle de Hodgkin-Huxley (HH) reste le plus complet et possède la particularité de pouvoir être paramétré à partir de paramètres biophysiques. Bien sûr, l’utilisation d’un modèle dépend de l’application visée. Un modèle présentant des caractéristiques biophysiques est plus adapté pour reproduire les maladies neurologiques.

Le modèle de Hodgkin-Huxley possède pourtant des limitations [BRETb 2015]. Le neurone est une cellule nerveuse possédant une morphologie complexe comprenant des ramifications dendritiques plus ou moins importantes et un axone. Cette morphologie induit un certain nombre de phénomènes non reproduits par le modèle de HH. D’autant plus que les maladies neurologiques affectent ces parties de cellules. Un type de modèle a vu le jour à la suite des théories de [RALL 1989] permettant de représenter les neurones sous un ensemble de compartiments. Les dendrites sont ainsi représentées par un nombre fini de câbles électriques afin d’augmenter la précision de la description des cellules nerveuses. Ce type de modèle est appelé modèle à plusieurs compartiments (multicompartmental en anglais).

L’équipe AS2N a par le passé réalisé des puces analogiques intégrant le modèle de Hodgkin-Huxley [SAIG 2011]. Cependant le coût de production ainsi que le manque de fiabilité dû à la complexité de paramétrage de ce type de système [GRASS 2011] ont entraîné une réflexion au sein de l’équipe sur l’utilisation de plateformes numériques telles que les Field Programmable Gate Array (FPGA). Une thèse a vu le jour sur l’implémentation du modèle d’Izhikevich sur FPGA [AMBR 2013], cependant, ce modèle possède des limitations. Les implémentations sur CPU ne permettent pas de simuler en temps réel et ont besoin de temps de simulation importants ne serait-ce que pour simuler quelques secondes d’activités d’un réseau de neurones. D’autres solutions sont portées sur GPU ou sur supercalculateurs. Cependant, elles ne permettent pas d’être utilisées à des fins d’interactions entre le vivant et l’artificiel et sont des technologies qui consomment beaucoup d’énergie [ALBA 2018].

La problématique de la thèse se situe sur l’implémentation et l’optimisation d’une plateforme permettant la simulation d’un réseau de neurones à des fins d’études de maladies neurologiques et sur ses applications aux expériences biohybrides à des fins biomédicales. Afin de répondre à cette problématique, nous proposons, dans un premier temps, une méthode optimale et flexible pour l’implémentation du modèle de Hodgkin-Huxley qui permettra la simulation d’une population relativement grande de neurones connectés. Dans un second temps, nous décrirons l’implémentation d’un modèle flexible à plusieurs compartiments. Pour implémenter ces modèles, une bibliothèque VHDL a été créée et a également

été utilisée dans d'autres applications.

Le premier chapitre a pour but de présenter de manière plus détaillée le contexte du sujet. Une première partie parlera de la morphologie biologique des neurones. La deuxième partie portera sur la transmission d'informations dans le cerveau. La troisième partie sera axée sur les maladies neurologiques, leurs afflictions et leurs effets potentiels sur le plan cellulaire. Cette partie traitera également des systèmes biohybrides. La partie suivante concernera les modèles de neurones en détaillant le modèle de Hodgkin-Huxley, ses limitations et les modèles à compartiments.

Le deuxième chapitre exposera, tout d'abord, le contexte technologique, à savoir, l'état de l'art sur les implémentations et la simulation de modèles de neurones utilisant diverses technologies. Ensuite, les propositions techniques permettant de répondre à la problématique seront détaillées. Cela inclura les propositions pour les implémentations de réseaux de neurones mais aussi pour les autres fonctionnalités telles que les synapses et la variabilité des influx nerveux dans les réseaux de neurones. Enfin, ce chapitre exposera des outils permettant de valider les modèles pour ensuite décrire les résultats.

Le troisième chapitre exposera la création d'une bibliothèque en VHDL optimisée pour l'ingénierie neuromorphique. La reconfiguration sur FPGA est une pratique permettant d'être au plus proche de la machine. L'avantage est la grande flexibilité de la technologie et les nombreuses possibilités d'optimisation. Cependant, l'inconvénient est la complexité d'utilisation des fonctions mathématiques et leurs coûts en implémentation. Ainsi, ce chapitre fera un état de l'art de solutions existantes avant de montrer les solutions proposées ainsi que leurs implémentations.

Le quatrième chapitre fera l'objet de la validation du système conçu par des applications expérimentales, d'une part, avec l'utilisation des réseaux de neurones exposés dans le deuxième chapitre et, d'autre part, avec l'utilisation de la bibliothèque VHDL décrite dans le troisième chapitre. La plateforme intégrant un réseau de neurones modulable a permis de réaliser des expériences biohybrides et d'afficher des résultats sur la connexion entre le vivant et l'artificiel. Cependant, les outils présentés dans le chapitre 3 ont aussi permis de cibler d'autres applications touchant à des sujets tels que l'optimisation combinatoire et les simulations de modèles biochimiques.

Chapitre 1

Du cerveau à la machine

1.1 Introduction

Ce chapitre a pour but de présenter le contexte de la thèse et de faire le lien avec la problématique. Il s'attelle à décrire les fonctions, l'anatomie des cellules nerveuses et le mécanisme de communication du système nerveux. Les maladies neurologiques seront abordées afin de décrire les possibilités que la simulation et les expériences biohybrides offrent dans le domaine biomédical. Ensuite, une partie sera consacrée aux différents modèles qui permettent la description du fonctionnement des cellules et les degrés de complexité nécessaires à l'étude, et spécifiquement appliqués aux maladies neurologiques. Enfin, ce chapitre exposera l'état de l'art des recherches permettant de simuler des réseaux de neurones sur diverses plateformes.

1.2 Morphologie du système nerveux

Le corps humain est composé de circuits complexes, permettant la transmission d'informations formant des structures anatomiques appelées système nerveux. Ce dernier est catégorisé sous deux appellations à savoir le système nerveux central (SNC), qui comprend l'encéphale et la moelle épinière, et le système nerveux périphérique (SNP) qui englobe les unités motrices et sensitives. Le système nerveux est composé, entre autres, de neurones et de synapses, cellules qui jouent un rôle important dans la transmission et le traitement d'informations.

1.2.1 Neurone

Un neurone est une cellule nerveuse qui possède une capacité de communication intercellulaire et de signalisation électrique. Pendant longtemps le nombre de 100 milliards de neurones a été propagé sans sources citées. [WILI 1988] a estimé, en moyenne, le nombre de neurones à environ 85 milliards chez l'humain, cependant, des études plus récentes estiment le cervelet à 101 milliards de neurones [ANDE 1992] et le cortex cérébral à 21 (moyenne sur 13 cerveaux d'hommes âgés de 57,4 ans) à 26 milliards (moyenne sur 18 cerveaux de femmes âgés de 65,1 ans) [PELV 2008] ce qui augmente le nombre moyen à plus de 120 milliards [SUZA 2009]. Le neurone possède une taille comprise entre 4 µm et plus de 100 µm de diamètre chez l'humain et a la particularité de posséder un noyau bloqué en interphase, il est donc incapable de se diviser. Pendant longtemps, on pensait que les neurones ne se renouvelaient plus. Or, il a été découvert que de nouveaux neurones naissaient, ce phénomène est appelé neurogenèse. Comme toutes cellules biologiques, le neurone est constitué d'un noyau et est délimité par une membrane plasmique qui permet de contenir le cytoplasme. Le neurone possède également deux types de prolongement nerveux qui sont les axones et les dendrites. Ces prolongements s'interfacent via des synapses permettant la communication intercellulaire par transmission de neurotransmetteurs (Figure 1.1.C).

Parmi les nombreux types de neurones, il existe, des motoneurones permettant d'assurer la transmission entre le système nerveux et les fibres musculaires, des neurones sensitifs transmettant des informations vers le SNC, des interneurones faisant l'interface entre les deux types cités précédemment et des neurones corticaux. Ces derniers seront principalement ceux qui nous intéresseront dans ce manuscrit pour leur rôle dans les maladies neurologiques. La morphologie de plusieurs neurones est représentée sur la figure 1.1. Morphologiquement. Les cellules nerveuses peuvent être catégorisées selon trois familles : les

cellules unipolaires, bipolaires et multipolaires (Figure1.1.A).

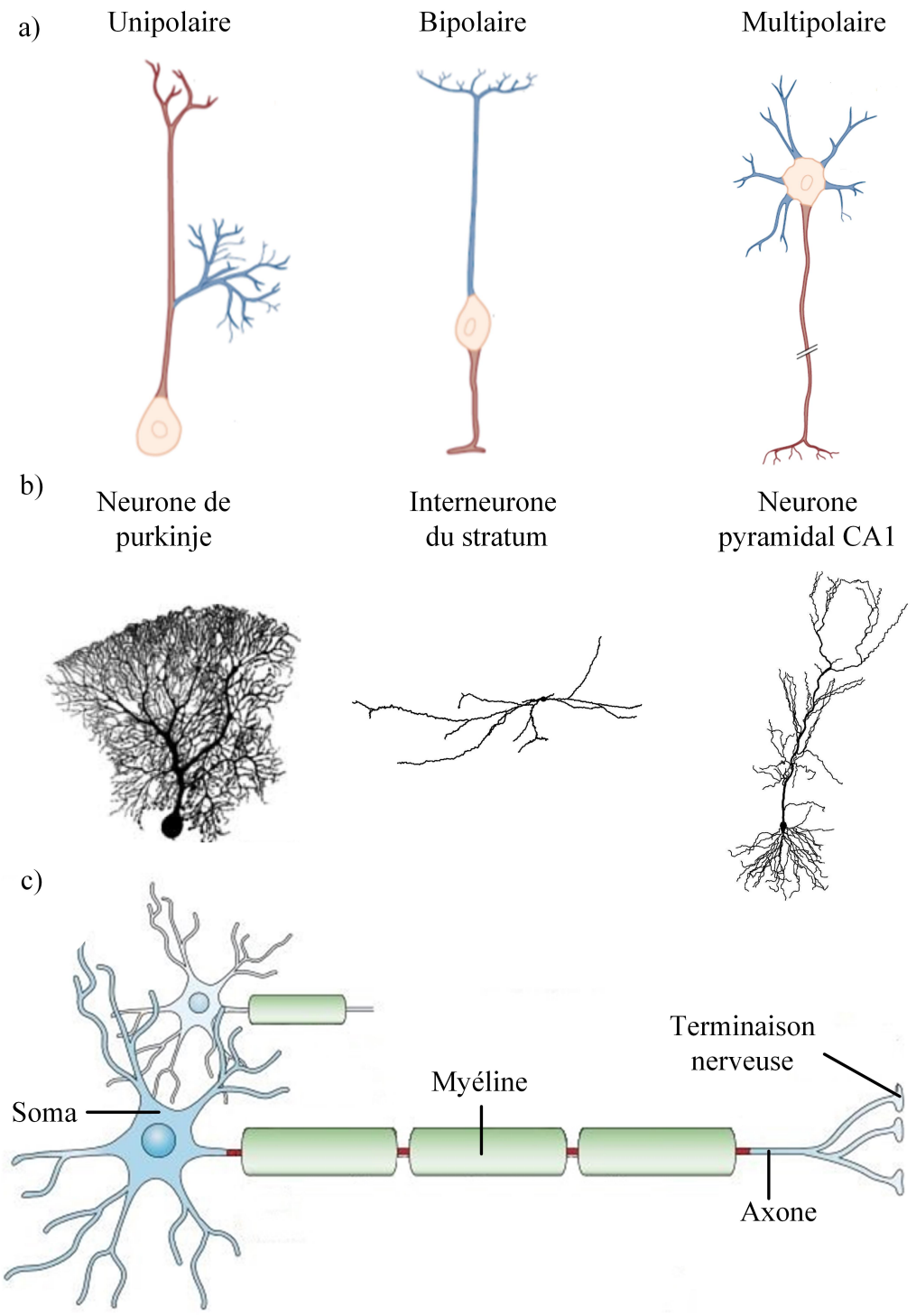

FIGURE 1.1 – a) Dessins d'une Cellule unipolaire, bipolaire et multipolaire issue de [KEND 1981] ; b) Différents neurones montrant différentes structures dendritiques : neurone de Purkinje se trouvant dans le cervelet [MCKA 2005], interneurone du stratum oriens dans l'hippocampe issu du site de Janelia.org, neurone pyramidal de la région ca1 de l'hippocampe reconstruit par [GOLD 2005], il s'agit bien évidemment de reconstructions 3D des cellules ; c) Anatomie du neurone récupérée et modifiée de [POLI 2003].

La diversité de ces cellules ne s'arrête pas à ces trois types, car il existe aussi une multitude de neurones dont la structure est différenciée par leurs ramifications dendritiques (Figure 1.B). Les patterns d'influx nerveux ainsi que la fonction des cellules seront traités plus loin dans le manuscrit.

1.2.2 Dendrites

Les dendrites sont des prolongements nerveux issus des neurones constituant une arborescence plus ou moins complexe dépendant du type de neurones. Ils interviennent dans la réception des potentiels d'action via les synapses et se différencient d'un axone par l'absence de myéline. Elles sont facilement confondues avec un axone si ce dernier est démyélinisé. Les dendrites peuvent également s'étendre, 10 à 20 fois la surface des neurones, depuis lesquels elles se ramifient [BMEL 2006]. L'importance des dendrites dans le fonctionnement du système nerveux fera l'objet d'une partie dans ce manuscrit. Ces ramifications reçoivent des informations sous forme de molécules chimiques appelées neurotransmetteurs, libérés par l'axone.

1.2.3 Axones

L'axone est un prolongement nerveux permettant la conduction à haute vitesse des influx nerveux aussi appelés potentiels d'action. Il est enroulé par plusieurs épaisseurs de protéines formées par des cellules gliales, oligodendrocytes dans le SNC et cellules de Schwann dans le SNP, qui créent ainsi des gaines appelées myéline qui permettent d'isoler électriquement la membrane et d'augmenter la vitesse de conduction électrique. L'axone est composé de plusieurs parties qui sont le segment initial de l'axone, les nœuds de Ranvier, les parties isolées par la myéline de l'axone et son terminal sur lequel se trouvent des boutons synaptiques permettant la libération des neurotransmetteurs au travers de synapses.

1.2.4 Synapses

L'axone se ramifie en plusieurs boutons synaptiques qui libèrent des composés chimiques de plusieurs natures à proximité des dendrites. Cette interface axone/dendrite est appelée synapse. Il en existe de deux types, les synapses électriques et les synapses chimiques qui sont présentes en majorité. Leur nombre par neurone varie selon le type de neurone et son emplacement dans le système nerveux.

1.2.5 Membrane plasmique

La membrane qui délimite le corps cellulaire est appelée membrane plasmique. Elle est constituée de deux types de protéines membranaires et d'une bicouche lipidique. Il existe d'une part les transporteurs actifs qui déplacent les ions dans le sens contraire de leurs gradients de concentration et, d'autre part, les canaux ioniques. Ces derniers permettent la perméabilité sélective de la cellule à certains ions ainsi que leur transfert. De nombreux types de canaux ioniques ont été trouvés à ce jour et possèdent tous des fonctions et caractéristiques différentes [TOLE 2004]. Cette composition lipide/protéine transmembranaire permet de séparer le milieu extra-cellulaire du milieu intracellulaire et d'imposer des gradients de concentrations ioniques. Ces deux milieux sont majoritairement composés d'ions potassium (K^+), d'ions sodium (Na^+), d'ions chlorure (Cl^-) et d'ions calcium (Ca^{2+}). La diffé-

rence de concentration ionique d'un milieu à l'autre joue un rôle important dans la création et la propagation de potentiels d'action. De manière générale, les ions sodium, chlorure et calcium sont plus importants dans le milieu extra-cellulaire que dans le milieu intracellulaire à l'inverse de l'ion potassium.

L'anatomie particulière et propre aux neurones permet de créer un environnement favorable à la transmission des influx nerveux qui sont considérés comme étant porteurs d'informations dans le cerveau.

1.3 Transmission de l'information

Les réseaux de neurones sont souvent décrits comme un circuit électrique complexe propageant des informations par signaux électriques. Le vocabulaire est emprunté aux domaines des télécommunications ou de l'électronique et est utilisé pour la modélisation des neurones.

1.3.1 Formation des potentiels d'action

La formation de potentiels d'action provient de mécanismes complexes issus des activités des canaux ioniques imposant des gradients différents de concentrations ioniques dans les milieux intra- et extra-cellulaires. Les différences de concentrations ioniques impliquent qu'il y a une différence de charges électriques et donc de potentiel. É tant donné qu'au repos, la membrane possède une plus grande perméabilité au potassium et que le pompage actif impose une concentration en potassium plus élevée dans le compartiment intracellulaire, nous pouvons calculer, grâce aux équations de Nernst, (équation 1.1), en prenant comme référence le milieu extra-cellulaire, le potentiel au repos.

$$E_X = \frac{RT}{zF} ln(\frac{[X]_{extra}}{[X]_{intra}}) \tag{1.1}$$

où E_X est le potentiel d'équilibre pour l'ion X, R la constante des gaz parfaits, T la température (en Kelvin), [X] la concentration de l'ion X, intra le compartiment intracellulaire et extra le compartiment extra-cellulaire.

Ainsi, connaissant les différences de concentrations ioniques entre les compartiments intra- et extra-cellulaire et en l'absence d'activité, le potentiel de la membrane est négatif. Les influx nerveux sont des signaux électriques se propageant sous une forme particulière et sont appelés Potentiels d'Action (PA). La figure 1.2 décompose la formation d'un PA à travers cinq étapes :

1) Le potentiel de membrane au repos est négatif (Figure 1.2.1).

2) Lorsque la membrane reçoit une stimulation électrique, la perméabilité au sodium augmente, entraînant une phase de dépolarisation ce qui a pour effet d'augmenter la tension de membrane (Figure 1.2.2).

3) Si la stimulation persiste, les canaux ioniques perméables au sodium s'activent laissant pénétrer les ions Na+ dans le compartiment intracellulaire (Figure 1.2.3).

4) Après cette activation des canaux sodium, s'ensuit une phase de repolarisation caractérisée par un phénomène d'inactivation des canaux sodium ainsi que d'activation des canaux potassium et d'augmentation de la perméabilité de la membrane au potassium ce qui a pour effet de faire chuter la tension (Figure 1.2.4).

5) Enfin, une dernière phase appelé hyperpolarisation est créée par le transport élevé de potassium vers l'intérieur de la cellule. De ce fait, la tension sera, pendant un moment, inférieure à la tension de repos (figure 1.2.5). On peut noter qu'il y a, pendant cette phase, une période dite réfractaire pendant laquelle aucun autre potentiel d'action ne pourra être formé.

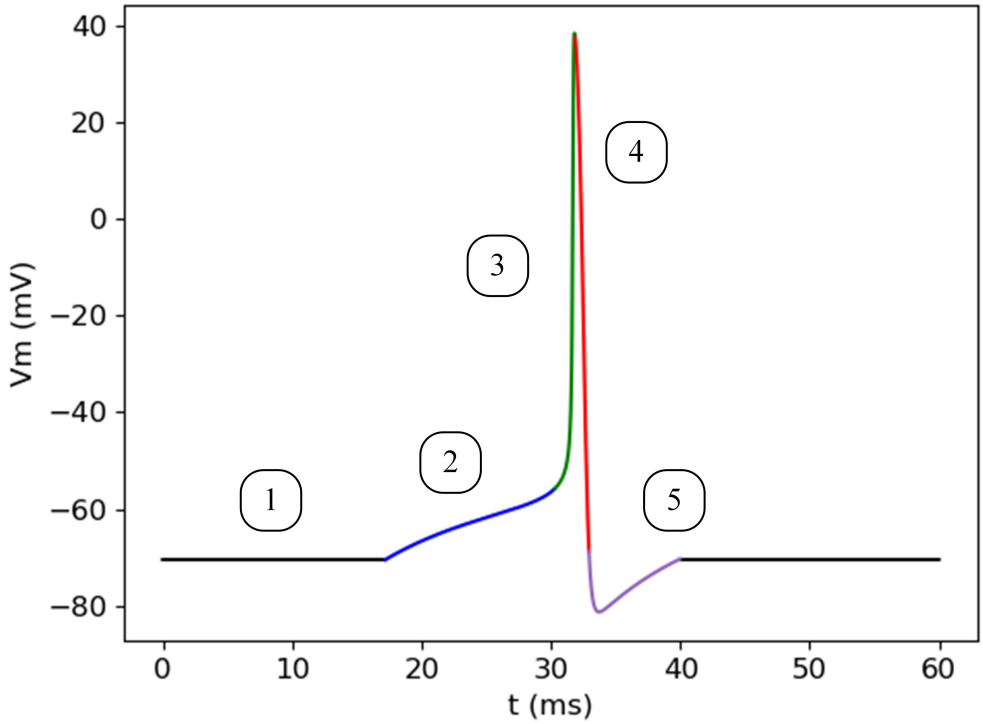

FIGURE 1.2 – Formation d'un potentiel d'action : 1) Le potentiel de membrane est à son potentiel de repos (−65 à −75 mv) ; 2) Il y a dépolarisation lors de la réception d'une stimulation ; 3) Si la stimulation persiste, l'ouverture des canaux sodium provoque un pic de tension ; 4) Les canaux sodium se referment et les canaux potassium s'activent ; 5) La perméabilité du potassium étant encore élevée, une phase d'hyperpolarisation survient, car la tension tend vers le potentiel d'équilibre du potassium. La tension de membrane retourne ensuite vers son potentiel de repos.

1.3.2 Transmission des influx nerveux

Nous avons précédemment expliqué que l'axone est un prolongement nerveux spécialisé dans la conduction de signaux électriques. Pourtant, cette conduction s'opère seulement lorsque l'axone est myélinisé et que le stimulus dépasse un certain seuil. De plus, elle n'est pas seule responsable de la transmission d'informations. Une autre structure biologique appelée synapse participe à la communication. Elle possède un mécanisme complexe

permettant le fonctionnement du système nerveux et de l'apprentissage.

Propagation dans l'axone

Lors d'une stimulation, si la dépolarisation n'est pas assez forte, il n'y a pas de déclenchement d'influx nerveux et le signal s'atténue au fil de l'axone comme sur la figure 1.3.a) qui représente l'expérience menée par [HODG 1936]. Cette expérience permet aussi d'observer que lorsque la membrane est passive elle fonctionne comme un condensateur. Contrairement à l'expérience précédente, dans le cas où l'axone reçoit une stimulation suffisante pour déclencher un potentiel d'action, l'axone propage l'influx nerveux de manière active sans atténuation (Figure 1.3.b). Un axone myélinisé propage les influx nerveux par conduction saltatoire. Il s'agit d'un mode de conduction particulier, les potentiels d'action sont régénérés au niveau des nœuds de Ranvier. Lorsque les canaux ioniques reçoivent un signal électrique, ils réagissent en réalisant le procédé expliqué précédemment pour former un PA. La propagation est unidirectionnelle, car la section de la membrane qui est en amont est en période réfractaire et ne peut ainsi plus générer de PA. Lorsqu'il y a dépolarisation sur un nœud de Ranvier, le courant s'étend jusqu'au nœud situé en aval de la membrane, jusqu'à activer les canaux du prochain nœud qui va à son tour former un PA.

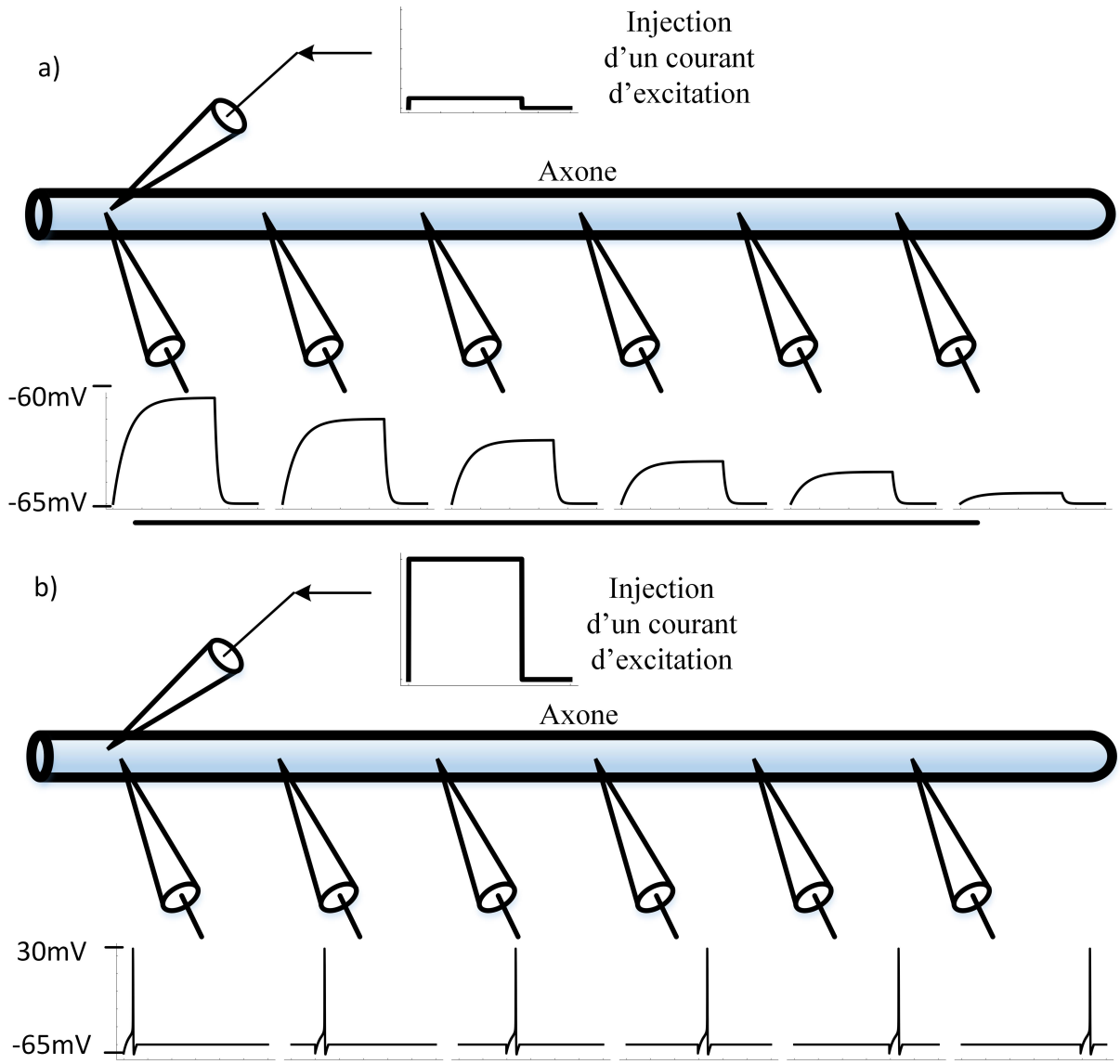

FIGURE 1.3 – a) Une électrode injecte du courant dans l'axone, plusieurs électrodes d'enregistrement sont placées à plusieurs endroits, le courant injecté est trop faible pour générer un potentiel d'action, la conduction reste passive ; b) Même expérience que a) à la différence que le courant injecté est suffisant pour créer un potentiel d'action, la propagation active ne provoque pas d'atténuations.

Les synapses

Une fois que le PA traverse l'axone jusqu'au terminal axonal, le signal passe par la synapse qui interface l'axone avec la dendrite. Á ce jour, deux types de synapses ont été distingués dans le système nerveux, les synapses électriques et les synapses chimiques. Avant d'aller plus loin, il est généralement admis que la source de l'information, à savoir de l'influx nerveux, est appelé présynaptique et que le neurone récepteur est appelée postsynaptique. Dans le cas d'une synapse électrique, les canaux membranaires post et présynaptiques sont au contact et se font face. Cet appareillage entre les canaux est appelé jonction communicante et c'est par là que passent les différents ions. Le Post Synaptic Potential (PSP) qui est le potentiel mesuré sur le neurone postsynaptique présente un délai moins court que dans le

cas de la synapse chimique. Ce dernier enclenche un processus biochimique complexe lors de la réception d'un potentiel d'action (Figure 1.4). Tout d'abord, des neurotransmetteurs sont stockés dans des vésicules qui vont fusionner avec la membrane lors de la réception d'un potentiel d'action (Figure 1.4.a). Le PA aura pour effet l'activation des canaux ioniques et l'augmentation du calcium dans le bouton synaptique ce qui provoque la fusion de la vésicule avec ce dernier (Figure 1.4.b). Les neurotransmetteurs sont ensuite libérés et liés, pour la plupart, aux récepteurs présents sur la membrane postsynaptique. Les canaux de cette dernière s'activent pour laisser entrer les ions et ainsi générer un courant postsynaptique (figure 1.4.c).

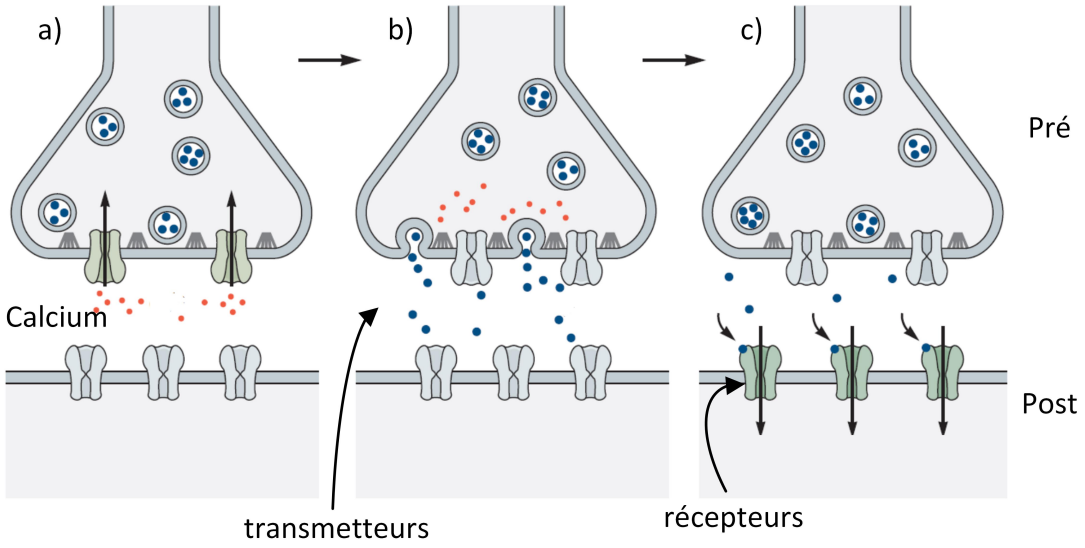

FIGURE 1.4 – Schéma d'une synapse récupéré et modifié de [KEND 1981] ; a) Les neurotransmetteurs sont capturés dans des vésicules synaptiques. Un potentiel d'action survient et provoque la dépolarisation du terminal laissant ainsi pénétrer des ions calcium ; b) Les vésicules fusionnent avec la membrane présynaptique et les neurotransmetteurs sont libérés ; c) Les neurotransmetteurs se lient aux récepteurs et génèrent un courant postsynaptique.

Les synapses sont catégorisées par deux fonctions. Elles peuvent être inhibitrices ou excitatrices, bien qu'il s'agisse d'une version simplifiée de ce qu'il se passe dans les systèmes biologiques. On appelle Excitatory Post Synaptic Potential (EPSP) le potentiel électrique, permettant l'excitation, qui est présent sur la membrane postsynaptique et Inhibitory Post Synaptic Potential (IPSP), le potentiel qui permet l'inhibition. Ce mécanisme d'excitation et d'inhibition est représenté sur la figure 1.5.

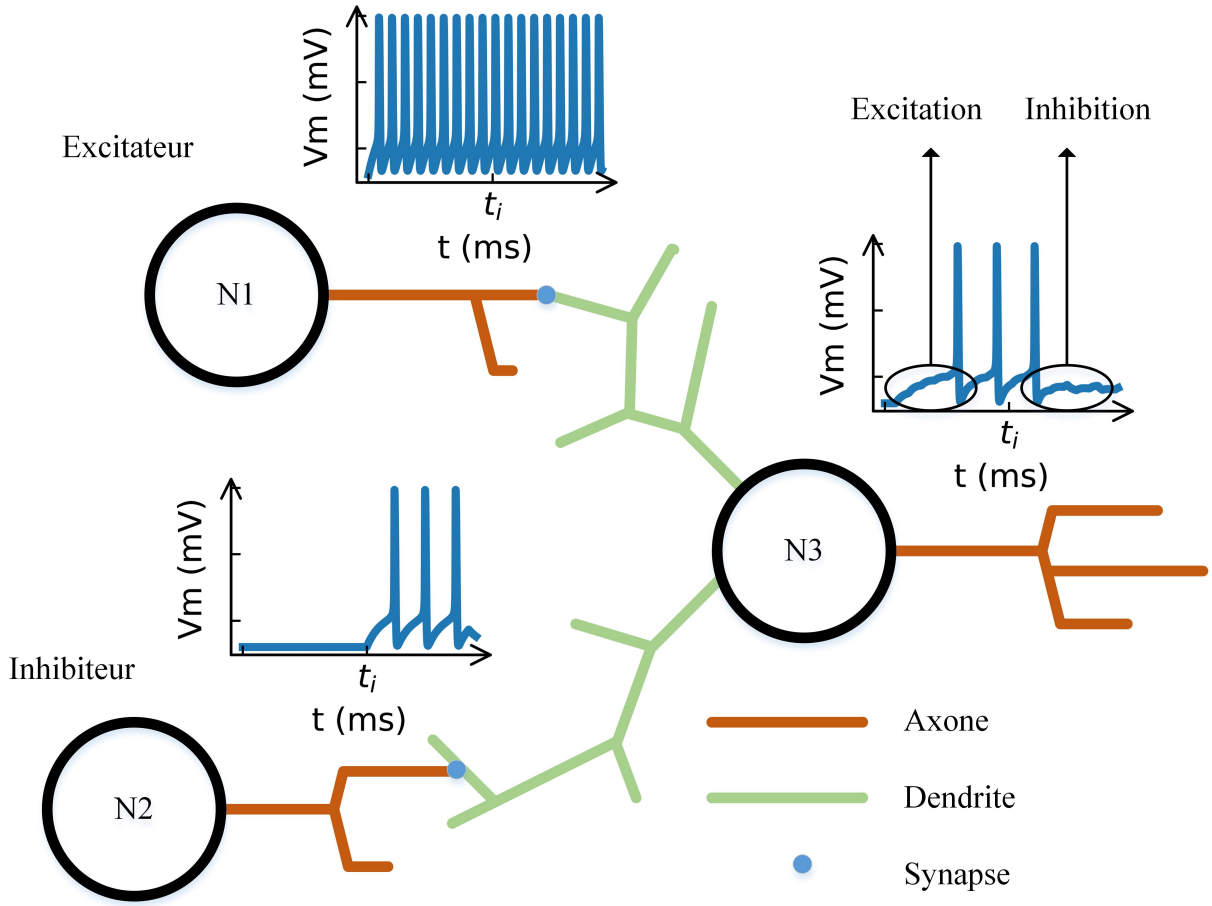

FIGURE 1.5 – Deux neurones transmettent une information à un troisième. Le premier possède une synapse excitatrice et le deuxième une synapse inhibitrice qui génère des PA au temps ti. Le potentiel postsynaptique subit une dépolarisation lors de la réception de l'EPSP et une repolarisation lors de la réception de l'IPSP.

La figure 1.5 montre un exemple d'EPSP et d'IPSP visibles sur le neurone 3 après transmissions de potentiels d'action d'un neurone 1 excitateur et d'un neurone 2 inhibiteur. La fréquence du potentiel d'action de la cellule excitatrice provoque une dépolarisation qui finit par déclencher des potentiels d'action. Ces derniers seront inhibés lorsque le neurone 2 commencera à transmettre.

Une autre particularité des synapses est leur plasticité. En effet, l'expérience montre que lorsqu'un *spike* (de l'anglais pique désignant les PA) intervient, la synapse serait susceptible de déclencher un autre potentiel d'action plus rapidement. On dit alors que la synapse possède un poids qui se renforce. [HEBB 1949] propose une théorie et dit sur le sujet : « »Lorsqu'un axone de la cellule A est suffisamment proche pour exciter B et participe de manière répétée ou persistante à son déclenchement, un processus de croissance ou un changement métabolique se produit dans une ou les deux cellules, de telle sorte que l'efficacité de A, en tant que cellule activant B, soit augmentée. (Traduit de [HEBB 1949].)

L'expérience montre aussi que si le neurone présynaptique déclenche (génère un PA) avant le neurone postsynaptique, il y a alors une augmentation du poids synaptique (potentialisation). Si le contraire arrive, on parle de dépression de la synapse [BLISS 1973]. Cette

association de phénomènes est appelée Spike Timing Dependent Plasticity (STDP) et jouerait un rôle important dans l'apprentissage. La théorie différencie deux types de STDP. Il y aurait un phénomène de plasticité à court terme et un phénomène à long terme. Le premier est caractérisé par la potentiation ou la depression des synapses sur un temps réduit (de la seconde à la minute) contrairement au deuxième qui maintiendrait sa fonction plus longtemps. Alors que la majorité des publications s'accordent à utiliser ce principe, d'autres études [BONO 2017] suggéreraient que la STDP n'est pas la seule loi d'apprentissage. Cependant, ce manuscrit n'a pas pour but d'expliquer les mécanismes complexes de l'apprentissage dans le détail et se contentera des systèmes de transmission dans des réseaux de neurones limités en nombre.

La transmission synaptique est la représentation la plus utilisée de la transmission d'informations. Cependant, certaines études suggèrent qu'il est possible que l'information puisse aussi passer par le milieu extra-cellulaire [MICH 1998]. Certaines synapses seraient « ouvertes » et libéreraient des neurotransmetteurs dans le milieu extra-cellulaire afin d'être transmis à une plus grande population de neurones. Ce mode de transmission serait plus important chez les personnes atteintes de maladies neurodégénératives telles que Parkinson. [FABE 1989] suggèrent un autre mécanisme de communication. Ils expliquent que les activités électriques du système nerveux produit des champs électriques qui affecteraient les cellules voisines jusqu'à provoquer des PSP. Il est bon de noter qu'il se pourrait que la synapse ne soit pas le seul moyen de communication que le cerveau possède cependant, ce type de transmission n'est pas encore bien compris ni modélisé. De plus, il n'est pour le moment pas pertinent dans le cas de l'étude des maladies neurologiques.

La transmission d'un neurone à l'autre se fait via les synapses qui relaient les informations propagées par l'axone vers une dendrite qui sert d'intégrateur des signaux présynaptiques pour le neurone postsynaptique.

1.3.3 Variabilité

Lors de mesures de PA sur des neurones biologiques, il a été observé que d'une expérience à une autre les fréquences des trains de *spikes* étaient différentes [BRIT 1996]. Plusieurs facteurs seraient à l'origine du bruit dans les réseaux de neurones. [RENA 2014] catégorise l'origine du bruit par les « variabilités externes » et les « variabilités internes ». Le premier concernerait les générateur de bruits dont on ne contrôlerait ou ne connaîtrait pas l'origine et le second les sources présentes dans le cerveau. Il catégorise aussi les sources de variabilité contrôlables et celles incontrôlables. Parmi les premières, on peut citer, lors d'expériences sur des animaux, les stimuli sensoriels. Dans les seconds cas, les fluctuations dues à la thermodynamique où les activités chaotiques dans le système nerveux peuvent faire partie des sources incontrôlables lors d'expériences. Dans le cadre de l'étude des maladies et de la simulation de réseaux de neurones, seules les sources présentes dans le cerveau pourront être simulées, et de ce fait, ce sont les seules que nous prendrons en compte. [MANW 1999] explique que le bruit dans les réseaux de neurones est dû au bruit thermique causé par la résistance des membranes passives, à l'ouverture et à la fermeture des canaux ioniques voltage-dépendants et à l'activité aléatoire des synapses et plus précisément au mécanisme de libération des neurotransmetteurs via les vésicules synaptiques. Une autre source de bruit aurait pour origine les activités des réseaux de neurones. Certaines simulations de populations de neurones auraient généré des activités chaotiques [MONTE 2012]. Á

la suite des observations de la variabilité dans le vivant, une autre discussion dans le monde neuroscientifique a été ouverte, à savoir, la sensibilité au bruit. Ainsi, [LONDO 2010] a montré que les réseaux de neurones corticaux possédaient une faible sensibilité au bruit. Une faible variabilité aurait tendance à induire de grandes variations dans les activités. Aussi, les fluctuations du neurone seul ne suffiraient pas à changer la dynamique de réseau. Ces expériences et arguments on fait développer une forte croyance sur le fait que le cerveau « chiffrerait », mot importé du vocabulaire de cryptologie informatique, l'information à travers un nombre moyen d'influx nerveux ou *spikes*. Cependant, une autre théorie appuierait que le timing des *spikes* serait important. Pour finir sur la présence d'activités spontanées dans le système nerveux, des études ont montré qu'elles ont une importance sur l'attention et la mémoire [FOX 2006].

La présence du bruit cause de nombreuses fluctuations dans le système nerveux amenant la communauté scientifique à faire émerger de nouvelles théories et des débats sur le « codage » de l'information par le système nerveux. De plus, des études montrent des corrélations entre l'utilisation des fonctions cognitives telles que l'attention ou la mémoire avec la présence d'activités spontanées. Ainsi, l'importance du sujet de l'encodage et du décodage des informations dans le système nerveux fera naître plusieurs sujets de recherche.

1.3.4 Le codage des informations

Les potentiels d'action sont considérés comme la méthode de communication entre neurones et entre réseaux de neurones. La variabilité des neurones ainsi que le manque de répétabilité des expériences ont permis à deux grandes théories de voir le jour afin d'expliquer le « codage » des potentiels d'action dans le cerveau. Il y a d'un côté le « *rate coding* » pour lequel le nombre de *spikes* est important et le « *temporal coding* » pour lequel le timing des PA est important.

Codage fréquentiel

Le concept de « *rate* » est apparu avec les expériences de [ADRI 1926] sur les grenouilles et des récepteurs nerveux. Les enregistrements montrent qu'en augmentant la pression appliquée sur le muscle, la fréquence de *spikes* augmente (Figure 1.6). Dans cette expérience, le stimulus aurait un effet important sur la fréquence de décharge des PA.

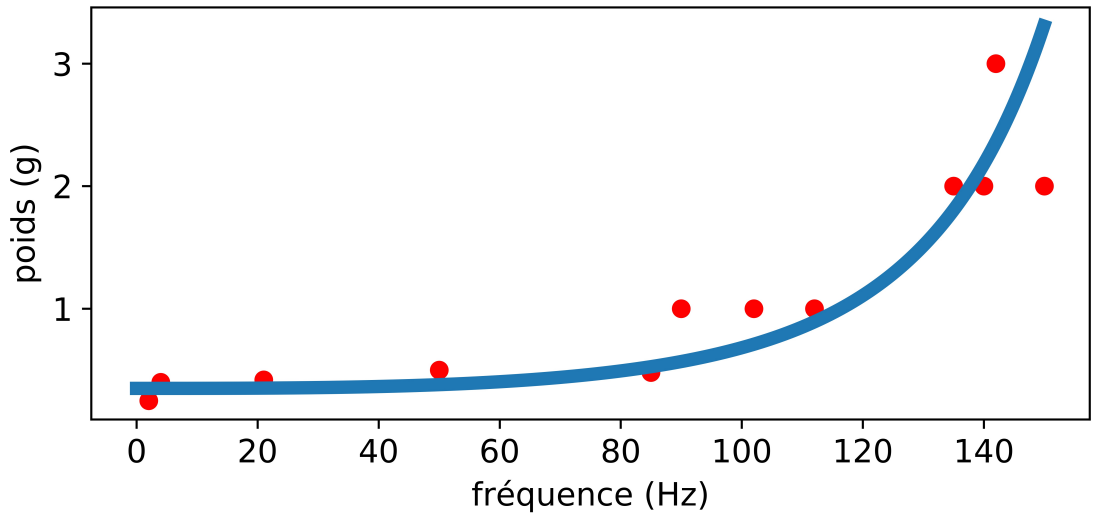

FIGURE 1.6 – Poids appliqué sur un muscle de grenouille en fonction de la fréquence de *spikes* mesurée sur des récepteurs nerveux, ces résultats sont reproduits depuis [ADRI 1926].

[BRETb 2015] décrit le concept de « *rate* » en codage de neurones comme étant une construction mathématique abstraite de calculs sur une infinité de *spikes* et sur une quantité moyenne définie sur le timing des *spikes*. La variabilité des neurones ainsi que le manque de répétabilité des expériences a amené à réfuter le concept basé sur le « *timing* » des potentiels d'action.

Cependant, [ABEL 1993], [THOR, 1996] et [ABEL, 2001] ont observé que le comportement des neurones du cortex visuel aurait une précision accrue aux timings de *spikes*. Ainsi, un autre concept majeur est né qui est basé sur les *spikes* individuels et leurs *timings*.

Codage temporel

Ce concept se fonde sur la caractérisation d'évènements discrets du temps de libération des vésicules synaptiques sur les terminaux des axones. Les *spikes* individuels sont associés à des mesures quantifiables et sont considérés comme la base de la communication entre neurones. Les théories fondées sur le timing de PA ne réfutent pas l'importance de la fréquence de ces derniers, en revanche, ces théories ne partent pas du principe qu'il est le paramètre essentiel à la communication.

Ces différentes théories amènent à réfléchir sur plusieurs points. Dans le cas où l'on souhaiterait faire communiquer la machine avec le vivant, doit-on se consacrer à reproduire une forme de *spike* au plus proche de la biologie afin de respecter le *timing* des PA ou doit-on envoyer des stimulations dont la fréquence et le nombre moyen d'impulsions suffirait à déclencher une réponse du réseau en ignorant la dimension *timing* et forme du PA. En effet, comme cité précédemment, les axones, comme les dendrites sont capables de générer un potentiel d'action à partir d'un stimulus tant que ce dernier permet au potentiel de membrane de dépasser le seuil de déclenchement de la membrane (rhéobase) pour former un PA. La question s'applique aussi dans le cas de la simulation d'un réseau de neurones. Une autre piste de réflexion sur le même sujet porte sur l'étude des maladies neurologiques. La modification des états biologiques des cellules, induite par les troubles neurologiques,

dans le système nerveux implique un changement dans la transmission des influx nerveux. Ce changement se traduit par une déformation du potentiel d'action mais aussi de sa fréquence de *spiking*.

1.4 Maladies neurologiques

Les maladies neurodégénératives ou neurologiques provoquent la perte de neurones et/ou de cellules gliales. Chaque année, l'Organisation Mondiale de la Santé (OMS) recense une augmentation des maladies neurologiques. Une des prévisions concerne la démence qui « triplera au cours des trente prochaines années » (Communiqué de presse du 7 décembre 2017 de l'OMS). D'une maladie à une autre, les effets sur les patients et sur les cellules sont différents. [OLLE 2006] résume les effets des principales maladies neurodégénératives sur les cellules biologiques. Ce manuscrit n'a pas pour but d'être un traité de médecine ni de prétendre décrire les maladies neurologiques en prenant en compte tous les détails ni d'ailleurs de prendre en compte toutes les maladies neurodégénératives. Cependant, il résume brièvement les effets de certaines maladies sur un patient humain ainsi que les potentiels effets sur la biologie qui pourraient être pertinents dans une simulation.

1.4.1 Un aperçu des maladies neurodégénératives

Maladie de Huntington

La maladie de Huntington affecte les neurones de projection dans le striatum, une structure nerveuse sous le cortex. Les neurones de cette région possèdent des axones plus longs que la moyenne et peuvent aller jusqu'à 1 mètre ce qui permet de communiquer avec plusieurs parties du système nerveux. Le striatum serait aussi une partie atteinte par la maladie de Parkinson. Les symptômes sont l'apparition progressive de mouvements spontanés involontaires et la perte progressive des capacités cognitives. La maladie de Huntington possède la particularité d'atteindre des neurones dont les axones possèdent une certaine taille. De ce fait, ce paramètre pourra être pris en compte lors des simulations.

Sclérose Latérale Amyotrophique (SLA) ou maladie de Charcot

Les précédentes maladies s'axaient sur des neurones corticaux, or, la Sclérose Latéral Amyotrophique (SLA) ou maladie de Charcot entraîne un dysfonctionnement des neurones moteurs dans la moelle épinière, le cortex cérébral et le tronc cérébral. Les symptômes sont l'apparition de faiblesses musculaires allant jusqu'à la paralysie, puis la mort. Les neurones corticaux restent les neurones les plus complexes. Il sera donc plus convenu de s'axer, lors des simulations, sur des neurones corticaux dont les modèles pourront être adaptés au besoin afin de devenir des neurones moteurs.

Maladie d'Alzeihmer

La maladie d'Alzheimer entraîne une perte des neurones mais aussi des synapses impliquant des dysfonctionnements dans l'amygdale, l'hippocampe et d'autres zones corticales. Cette maladie affecte progressivement les capacités cognitives et la mémoire. Cela appuie le besoin pour les simulations d'utiliser des modèles de neurones corticaux.

Sclérose en plaques

La sclérose en plaques, maladie affectant le système nerveux central affecte le mécanisme de propagation des potentiels d'action à travers l'axone soit, plus précisément, les canaux ioniques présents dans les nœuds de Ranvier. Elle a tendance à induire une destruction de la myéline entraînant la perte d'axones. L'implication de la vitesse de conduction de l'axone (principalement liée à la démyélinisation) et/ou de la destruction de ces conducteurs d'influx nerveux peut constituer des paramètres possibles à prendre en compte dans la simulation de réseaux de neurones.

Autres maladies neurologiques

Plusieurs autres déficiences ou anomalies du système nerveux peuvent entraîner diverses pathologies. Parmi elles, on peut citer les anomalies dues aux canaux ioniques. [TOLE 2004] ont montré le caractère diversifié des canaux ioniques voltage-dépendants. Ils possèdent tous des fonctions et des emplacements spécifiques faisant du neurone ce qu'il est. Une malformation ou une mauvaise localisation pourrait affecter la communication dans un réseau de neurones selon [HELE 2006]. De plus, il est possible de naître avec des malformations ou disjonctions de ces canaux ioniques, nommées canalopathies génétiques, qui pourraient survenir dans tout le système nerveux et créer des troubles neurodégénératifs [SPIL 2016].

Un autre exemple de facteurs possibles de maladies du système nerveux est la taille des neurones. Les expériences de [KERN 1981] ont montré l'effet de la taille des neurones sur la conduction, soit la rapidité de la transmission des influx nerveux dans les axones. Cette augmentation de la taille des neurones serait un des symptômes de la sclérose tubéreuse complexe (TSC en anglais) ou Sclérose Tubéreuse de Bourneville [KESS 2010]. Cette maladie entrainerait aussi des dysfonctionnements au niveau des synapses [BATE 2013] et des cellules gliales [COST 2016].

Ces symptômes entraînent principalement des modifications et dysfonctionnements dus aux canaux ioniques, à la morphologie des neurones (donc les dendrites et la taille des neurones) et des cellules gliales. En ajoutant les effets des autres maladies citées précédemment, on peut en déduire qu'il est aussi important de prendre en compte les autres parties de cellules telles que les synapses, l'axone et les dendrites.

1.4.2 Étude des maladies neurologiques

L'augmentation des maladies neurodégénératives a amené les scientifiques à reconsidérer la manière d'étudier les cellules biologiques et le traitement des maladies. Au même moment, les évolutions de l'ingénierie et des neurosciences ont permis la création de nouveaux domaines tels que les neurosciences computationnelles ou l'ingénierie neuromorphique.

Plusieurs disciplines s'attellent à étudier les maladies neurodégénératives de plusieurs manières telles que l'étude des mécanismes qui entraînent les maladies, les outils de diagnostic ou des moyens de traitement.

Simulation

L'émergence des neurosciences, et plus particulièrement des neurosciences computationnelles ces dernières années, a permis l'étude et la création d'outils de simulation de modèles de systèmes biologiques. Ces outils ont entraîné la création de théories comme celles citées dans la partie « Transmission d'informations » et la découverte de mécanismes au sein du système nerveux [RECC 1999]. Le résumé des maladies dans la section précédente a permis de déterminer que les maladies neurodégénératives affectent certains types de neurones, des canaux ioniques spécifiques, elles peuvent atteindre la myéline voire la destruction de l'axone et peuvent également affecter la taille des neurones ou leur destruction.

Des équipes de recherche à travers le monde étudient les modèles de neurones biodétaillés afin d'avancer dans la compréhension des maladies neurodégénératives. Les modèles à plusieurs compartiments (qui seront expliqués plus tard dans ce manuscrit) sont utilisés, car ils prennent en compte les caractéristiques des maladies citées précédemment. Par exemple, [MART 2013] utilise un modèle à plusieurs compartiments pour simuler et prédire les effets d'hyperexcitabilité des motoneurones dans le cas de la SLA. La majorité des modèles de neurones s'attellent à reproduire et prédire les activités électriques des cellules nerveuses. D'autres chercheurs ont voulu simuler les réactions biochimiques qui se produisent dans le système nerveux telles que l'Adénosine-Triphosphate (ATP) pour étudier leurs effets sur les neurones [LEMA 2014]. L'ATP est une molécule qui servirait à fournir de l'énergie pour les réactions biochimiques des cellules. Dans le cadre de la SLA, la dégénérescence des neurones moteurs pourrait être due à niveau insuffisant d'énergie.

***Brain Machine Interface* (BMI)**

L'étude des mécanismes du système nerveux et des maladies neurodégénératives peut être faite en réalisant des expériences biohybrides vivant/artificiel. La connexion entre des systèmes artificiels et des cellules permettrait de « piloter » et de comprendre le vivant. Dans le cas de puces électroniques, on parle alors de systèmes neuromorphiques ou d'interface cerveau-machine (BMI pour Brain Machine Interface en anglais). Plusieurs systèmes tels que ceux présentés par [LEBED, 2009], [HOCH, 2012] et [BONI 2013] se sont concentrés sur les interactions entre des puces neuronales permettant de reproduire les activités électriques des réseaux de neurones et des cellules biologiques. Le but est de remplacer des zones endommagées du cerveau par ces systèmes. Pour atteindre ce but, il faut, d'une part, acquérir et interpréter les mesures prises sur le cerveau et, ensuite, renvoyer une réponse interprétable par le système nerveux. Ce type de systèmes est appelé *closed-loop devices* (appareils en boucle fermée) et possède une importance capitale pour la recherche sur les neuroprothèses [LEVI 2018]. [POTT 2014] montre les innovations sur ce type de systèmes et présente le *Micro Electrode Array* (MEA) comme un outil pour mesurer les influx nerveux issus de neurones. [BROC 2017] présente d'autres types de systèmes d'enregistrement tels que le *patch clamp*, les électrodes intracellulaires, les électrodes de surface et les électroencéphalogrammes (EEG). Ces plateformes d'enregistrement sont classées du plus proche du potentiel d'action (le *patch clamp*) au plus éloigné (l'EEG) prenant en compte l'enregistrement des fréquences des ondes produites par le cerveau. Il présente aussi l'état de l'art de la recherche sur des travaux portant sur les BMI et la faisabilité de l'interaction entre le vivant et l'artificiel. Plusieurs travaux montrent qu'il est en effet possible d'interagir avec la biologie en utilisant des neurones en silicium. [JOUC 2016] et [CHOU 2014] décrivent plusieurs

travaux sur certains de ces systèmes neuro-biohybrides.

Ces travaux permettraient d'améliorer la recherche sur le biomédical et sur les troubles touchant le système nerveux. D'autres systèmes permettent d'interfacer le cerveau avec les machines [CARM 2003]. Un exemple concret peut être cité par les expériences de [VELL 2008] visant à permettre à un singe de se nourrir à partir d'un bras robotique en lui attachant les bras. Certains de ses neurones moteurs utilisés dans le contrôle du bras ont été connectés au bras artificiel. Cette expérience montre les possibilités d'améliorer et d'adapter les interfaces cerveau-machine aux êtres humains. Il serait par exemple possible de rétablir le sens du toucher par technologie BMI [ODOH 2012]. Ce type d'interfaces permettrait également de rétablir les fonctions motrices chez les personnes atteintes de paralysie comme, par exemple, l'utilisation de BMI pour la réparation de lésions de la moelle épinière [JACK 2016] ou le rétablissement de la communication entre deux réseaux de neurones biologiques [BUCCE 2019]. D'autres systèmes permettent une rémission de certaines maladies par stimulation du système nerveux [HAMM 2007]. Les transplantations technologiques sont vues comme des alternatives aux traitements chimiques. Cependant, il existe aussi des méthodes utilisant les cellules-souches [OLLE 2006]. De nombreuses recherches sur le sujet sont aujourd'hui menées et permettraient le remplacement de cellules endommagées dans le système nerveux.

Les maladies neurologiques sont en augmentation à travers le monde. La complexité du système nerveux fait de ces troubles un mystère non résolu à ce jour. Les symptômes peuvent aller de la faiblesse musculaire à la mort en passant par une dégénérescence des capacités cognitives et motrices. Au niveau cellulaire, ces maladies peuvent affecter la morphologie des neurones, la myéline, le fonctionnement des canaux ioniques et/ou les synapses. Ces paramètres sont à prendre en compte pour la suite de l'étude. Des chercheurs utilisent des modèles de neurones afin de mieux comprendre le fonctionnement de cellules. Les modèles ont déjà fait leurs preuves en permettant la découverte de phénomènes biologiques. Ils reproduisent et prédisent les activités électriques des neurones mais ces derniers ne sont pas qu'influx nerveux et, dans le cas de maladies neurodégénératives, des chercheurs ont émis l'idée qu'il était possible de simuler les interactions biochimiques au sein du système nerveux. Une autre façon de voir les choses est par la communication hybride entre la machine et les cellules. Alors que certains tentent de communiquer par stimulation électrique ou chimique avec les neurones [NISHI 2019], d'autres proposent des traitements par stimulation de certaines zones du cerveau. Pour finir, ces recherches pourraient mener à des alternatives de guérison ou d'amélioration du système nerveux par neuroprothèses ou cellules-souches.

1.5 Modèle de neurones

La complexité des interactions physiques, chimiques et biologiques a fait l'objet de nombreuses réflexions. La compréhension de phénomènes a amené l'être humain à interpréter et prédire les observations par ce qu'on appelle des modèles. Dans le cadre de la biologie, et plus précisément dans les neurosciences et la description des influx nerveux, il existe une multitude de modèles différents par leur utilité, leur complexité, leur niveau de détails et le comportement qu'ils décrivent. Il y a une séparation entre deux grandes familles de réseaux de neurones. D'un côté, il y a les réseaux de neurones inspirés de la biologie permettant de

réaliser des calculs tels que la reconnaissance d'image, la classification ou le traitement de données. De l'autre, les réseaux de neurones biomimétiques qui sont les réseaux présentés dans ce manuscrit. Parmi les réseaux de neurones biomimétiques, il est possible de retrouver deux catégories qui sont d'un côté les modèles à compartiment unique et les modèles à compartiments multiples. Dans le premier cas, la dimension spatiale, à savoir la forme complexe des neurones, est ignorée. Ainsi, les *spikes* ne sont initiés ni dans l'axone, ni dans la dendrite. Quant à eux, les modèles à compartiments multiples représentent différentes parties des cellules en plus du soma comme l'axone et la dendrite. Les modèles à compartiments uniques sont le plus souvent utilisés car, dépendants des applications, il n'est pas nécessaire de prendre en compte la structure (axone et dendrite) du neurone. Néanmoins, cela ne limite pas ou peu le nombre de modèles existants. [BRETa 2015] différencie les modèles dits bioréalistes des modèles dits biodétaillés. Il fait aussi la distinction entre un modèle très détaillé et un modèle réaliste. Il considère que le modèle *Leaky Integrate and Fire* (LIF) est réaliste en comparaison d'autres modèles tels que les modèles *Integrate and Fire* (IF) ou *Quadratic Integrate and Fire* (QIF).

1.5.1 Hodgkin-Huxley : Un modèle bioréaliste ou biodétaillé ?

Alan Lloyd Hodgkin et Andrew Huxley, de l'université de Cambridge, sont des physiologistes et biophysiciens anglais dont les travaux ont été largement reconnus. Ils ont obtenu le prix Nobel de médecine en 1963. Ils ont émis des théories sur le mécanisme de formation de potentiels d'action et la mécanique des canaux ioniques. En 1952, une série d'articles est publiée à leurs noms dans lesquels ils présentent, entre autres, un ensemble d'équations mathématiques permettant de reproduire les activités électriques des neurones à partir d'une représentation de la membrane sous forme d'un circuit électrique. La figure 1.7 permet d'obtenir l'équation 1.2.

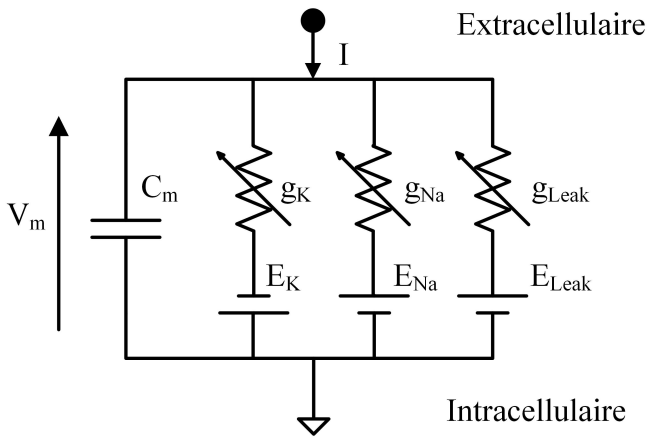

FIGURE 1.7 – Circuit équivalent selon Hodgkin et Huxley.

$$C_m \frac{dV_m}{dt} = I(t) - \sum I_{ion} \tag{1.2}$$

où C_m représente la capacité de membrane, V_m la tension de la membrane, I le courant entrant dans la membrane et I_{ion} les courants ioniques avec ion $\in$ {K, Na, Leak} représenté par les équations 1.3, 1.4 et 1.5.

$$I_K = \bar{g}_K n^4 (V_m - E_K) \tag{1.3}$$

$$I_N a = \bar{g}_{Na} m^3 h (V_m - E_N a) \tag{1.4}$$

$$I_l eak = \bar{g}_{leak} (V_m - E_l eak) \tag{1.5}$$

Avec g_{ion} la conductance maximale des canaux ioniques, n, m et h les fonctions d'activation et d'inactivation des canaux ioniques, V_m la tension de membrane et E_{ion} les potentiels d'équilibre des ions.

Hodgkin et Huxley ont déterminé les équations représentant l'activation et l'inactivation des canaux ioniques en fonction des termes voltage-dépendants alpha et bêta. [HODG 1952] détermine les fonctions alpha et bêta comme étant respectivement les fonctions de taux de transfert d'ions de l'extérieur vers l'intérieur et de l'intérieur vers l'extérieur. Ces fonctions sont représentées sur l'équation 1.6.

$$\frac{dx}{dt} = \alpha_x(V)(1-x) - \beta_x(V) \tag{1.6}$$

avec, x, la fonction d'activation ou d'inactivation des canaux ioniques, alpha et bêta les fonctions déterminant les taux de transferts d'ions, V la tension de membrane. Cette équation peut être réécrite par l'équation 1.7.

$$\frac{dx}{dt} = \frac{x_\infty - x}{\tau_x} \tag{1.7}$$

où,

$$x_\infty = \frac{\alpha_x(V)}{\alpha_x(V) + \beta_x(V)} \tag{1.8}$$

$$\tau_x = \frac{1}{\alpha_x(V) + \beta_x(V)} \tag{1.9}$$

Les paramètres constants sont la capacité de membrane, les conductances, les potentiels d'équilibre électrochimique des ions et les valeurs initiales (V, m, n, h).

1.5.2 Quel modèle choisir ?

Le choix d'un modèle dépend de l'application et du besoin. Dans le cadre de l'utilisation de réseaux de neurones pour des tâches de calculs telles que la reconnaissance d'image par exemple, il serait plus intéressant d'utiliser un modèle simple, peu coûteux et permettant d'intégrer un nombre de neurones plus élevé. En effet, plus le modèle est complexe et plus le temps de calcul sera long ou, dans le cas de circuit numérique ou analogique, le modèle sera lourd à implémenter et consommera de l'énergie. Or, ce manuscrit se concentre sur les modèles bioréalistes afin de pouvoir simuler les maladies et produire des influx nerveux en temps réel pour permettre des interactions avec des cellules biologiques dans le cadre d'expériences biohybrides. [BRETa 2015] compare les modèles IF, *exponential*, quadratique et Izhikevich afin de déterminer lequel serait le plus réaliste, c'est-à-dire, dont le comportement est le plus proche des cellules biologiques. Il affirme que le modèle d'Izhikevich s'éloigne du bio-réalisme et de sa capacité à prédire le comportement des neurones dans certaines conditions ce qui contredit les articles d'Izhikevich. Dans tous les cas, Hodgkin-Huxley est certes représenté comme le plus complexe mais aussi comme le plus complet des modèles

à compartiment unique.

Models	biophysically meaningful	tonic spiking	phasic spiking	tonic bursting	phasic bursting	mixed mode	spike frequency adaptation	class 1 excitable	class 2 excitable	spike latency	subthreshold oscillations	resonator	integrator	rebound spike	rebound burst	threshold variability	bistability	DAP	accommodation	inhibition-induced spiking	inhibition-induced bursting	chaos	# of FLOPS
integrate-and-fire	-	+	-	-	-	-	-	+	-	-	-	-	+	-	-	-	-	-	-	-	-	-	5
integrate-and-fire with adapt.	-	+	-	-	-	-	+	+	-	-	-	-	+	-	-	-	-	+	-	-	-	-	10
integrate-and-fire-or-burst	-	+	+		+	-	+	+	-	-	-	-	+	+	+	-	+	+	-	-	-		13
resonate-and-fire	-	+	+	-	-	-	-	+	+	-	+	+	+	+	-	-	+	+	+	-	-	+	10
quadratic integrate-and-fire	-	+	-	-	-	-	-	+	-	+	-	-	+	-	-	+	+	-	-	-	-	-	7
Izhikevich (2003)	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	13
FitzHugh-Nagumo	-	+	+	-		-	-	+	-	+	+	+	-	+	-	+	+	-	+	+	-	-	72
Hindmarsh-Rose	-	+	+	+			+	+	+	+	+	+	+	+	+	+	+	+	+	+		+	120
Morris-Lecar	+	+	+	-		-	-	+	+	+	+	+	+	+		+	+	-	+	+	-	-	600
Wilson	-	+	+	+			+	+	+	+	+	+	+	+	+	+		+	+				180
Hodgkin-Huxley	+	+	+	+			+	+	+	+	+	+	+	+	+	+	+	+	+	+		+	1200

FIGURE 1.8 – Tableau issu de [IZHI 2004] montrant les caractéristiques biophysiques de plusieurs modèles ainsi que leurs coûts de calcul en *FLOating-Points per Second* (FLOPS) ; Les coûts de calcul ont aujourd'hui évolué et cette unité ne vaut que pour les calculs utilisant un *Central Processing Unit* (CPU) et donc ne concerne pas les ASIC et les FPGA. En revanche, l'ordre de grandeur est représentatif de la complexité des modèles.

La plupart des systèmes utilisent un temps accéléré par rapport au temps biologique afin de simuler les réseaux de neurones d'où la précision du terme temps réel biologique. De plus, les maladies affectent surtout le comportement chimique du cerveau. Il est, de ce fait important de pouvoir conserver des paramètres biophysiques. [BRETa] sépare le terme bioréaliste du terme biodétaillé par le fait que les modèles décrivant le neurone en détail par des paramètres biophysiques ne veulent pas forcément dire qu'ils reproduisent les activités électriques avec précision. Ainsi, un modèle biodétaillé et bioréaliste à la fois sera favorisé afin de nous rapprocher d'un modèle biomimétique possédant une flexibilité qui permettrait d'adapter le modèle au besoin, à savoir, l'étude des maladies neurodégénératives. De plus, l'interaction interdisciplinaire avec des experts en neurosciences spécialisés en biologie ou biochimie en sera facilitée. [IZHI 2004] résume sur un tableau comparatif (figure 1.8), incluant le sien, en montrant que le modèle de Hodgkin-Huxley est le plus complet en matière de comportement, de prédictions et de paramètres biophysiques, ce qui en fait un modèle idéal pour l'étude des maladies neurodégénératives.

1.6 Modèles Biomimétiques

Les modèles de neurones peuvent représenter la cellule nerveuse dans sa plus grande simplicité à travers des représentations telles que le modèle *Integrate and Fire* (IF). Á l'opposé, le neurone peut être décrit avec plus de détails à travers des modèles comme celui de Hodgkin et Huxley. Ce dernier propose un set d'équations permettant de représenter la dynamique biochimique des canaux ioniques et ainsi de prédire la fréquence et l'amplitude de potentiels d'action à partir de paramètres biophysiques. Malgré la complexité des équations et le grand niveau de détail de ce modèle, il n'en reste pas moins une approximation et une simplification de ce que la nature produit. En effet, plusieurs parties des cellules telles que les dendrites, l'axone, la myéline, les nœuds de Ranvier ou les synapses (électriques ou chimiques) ne sont pas pris en compte malgré l'importance de leur présence. La simplification et l'approximation sont inévitables connaissant le nombre de facteurs agissant sur les cellules. De plus, il peut y avoir des facteurs génétiques sachant que les cellules ne seront pas les mêmes, dépendant de leur fonction et de leur emplacement dans le cerveau. Cette diversité amène au besoin de pouvoir classer les neurones.

1.6.1 Classification des neurones

Les neurones peuvent être classés selon différents critères. Ils peuvent être catégorisés par anatomie, par fonctionnement (excitateur ou inhibiteur), par pattern de potentiels d'action, par caractéristiques chimiques ou par expression de gênes. L'anatomie du neurone a déjà été traitée précédemment. Aussi, les caractéristiques chimiques ainsi que l'expression des gènes n'ont pas été estimées pertinentes. Nous nous concentrerons donc sur les *patterns* de *spikes* ainsi que les types de neurones.

Spikes

La classification des neurones peut se faire sur les patterns qu'il est possible de retrouver dans le système nerveux central. Plusieurs études ont permis de regrouper et de décrire plusieurs *patterns*. Parmi les *patterns* les plus importants, [GUTN 1990] en ont décrit trois qui sont représentés sur la figure 1.9. Les neurones de type *Regular Spiking* (RS) seraient les neurones les plus retrouvés dans le système nerveux. Lorsque la cellule est stimulée pendant une période assez longue, les enregistrements montrent une adaptation en fréquence. La figure 1.9.B montre que les premiers *spikes* sont plus rapides que les prochains. Les types *Fast Spiking* (FS), décrits par [MONC 1969], génèrent des trains de *spikes* plus rapides que les autres sans présenter d'adaptations de fréquence (Figure 1.9.A). Les types *Intrinsically bursting* (IB) se distinguent par les bouffées de potentiels d'action qu'ils génèrent. En effet, leurs potentiels d'action individuels ressemblent aux influx nerveux générés par les types RS mais ont la particularité de générer une bouffée de PA appelé *burst*. La figure 1.9.C montre une bouffée de PA lors des 250 premières millisecondes et une fréquence quasi constante par la suite. [IZHI 2003] présente plusieurs autres types de neurones dont le *Low-Threshold Spiking* (LTS) décrit par [GIBS 1999] et représenté sur la figure 1.9.D. Le LTS affiche une bouffée de PA lorsque le courant de stimulation passe d'une valeur négative à une valeur positive alors qu'il génère une fréquence de *spike* similaire au FS lorsque le courant de stimulation est positif et constant. [IZHI 2003] catégorise aussi les neurones en fonction de leur excitabilité. Les neurones de type FS et LTS sont inhibiteurs et les neurones de type RS et IB excitateurs.

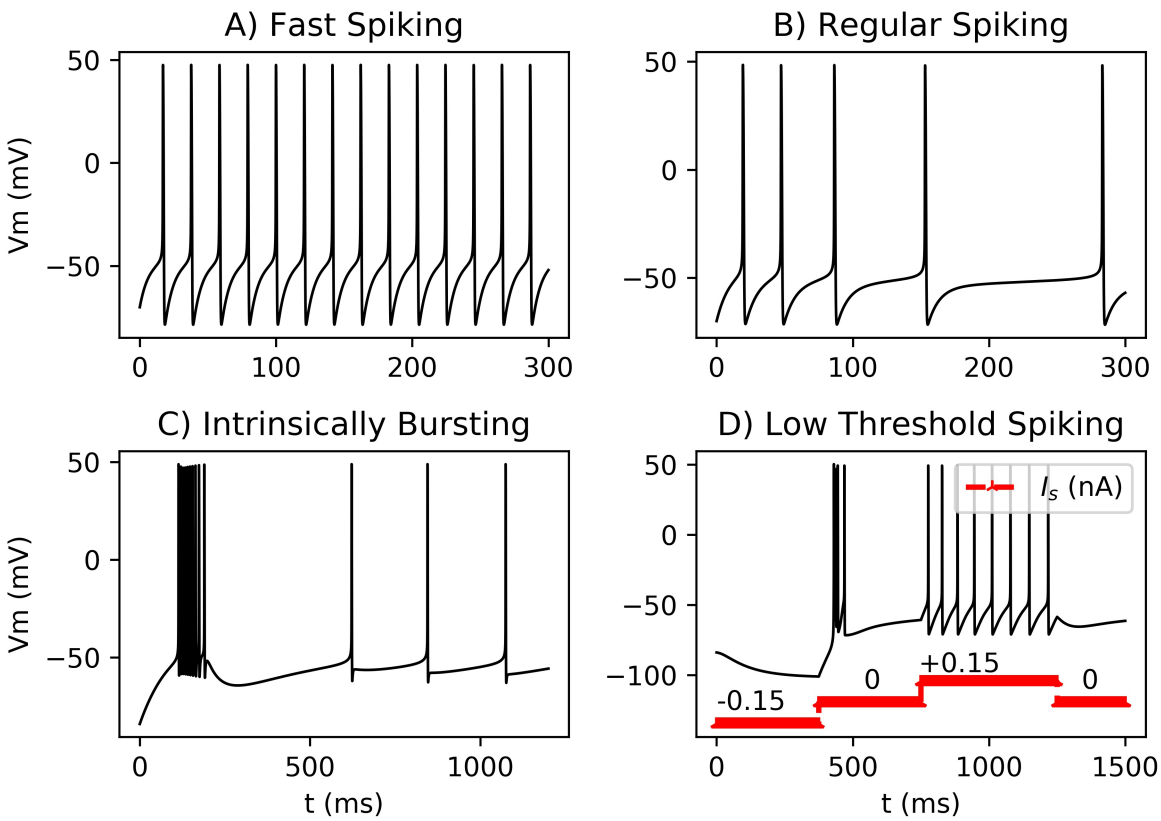

FIGURE 1.9 – Types de neurones par *patterns* de *spikes* simulés à partir des équations de [POSP 2008] : a) *Fast Spiking* (FS) ; B) *Regular Spiking* (RS) ; C) *Intrinsically Bursting* (IB) ; D) *Low Threshold spiking* (LTS).

Classification

Lors de ses expériences, [HODG 1948] répertorie ses observations concernant des différences dans les trains de *spikes* lors de même stimulus sur plusieurs neurones différents. Il classifie, à partir de ses résultats, les neurones en trois catégories. La différence entre les classes de neurones est représentée au travers de courbes F-I sur lesquelles le phénomène différenciateur apparaît (figure 1.10).

Les neurones peuvent être représentés par leur excitabilité. Pour être plus précis, il est possible de représenter graphiquement la réponse en fréquence de *spike* (en Hz) d'un neurone en fonction du courant d'excitation (généralement en pA) injecté sur le neurone. Ce type de graphique est appelé « courbe F-I » pour Fréquence-Intensité.

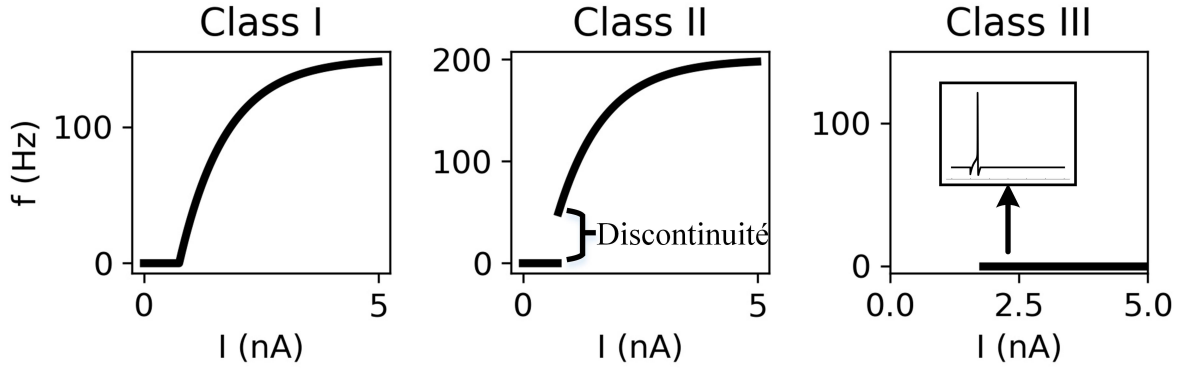

FIGURE 1.10 – La courbe F-I représente la stimulation du neurone en fonction de la fréquence de *spikes*. Cette figure est inspirée de [PRES 2014]. Les chiffres ne représentent pas de valeurs exactes mais un ordre de grandeur : A) Neurone de classe I, le neurone continue de répondre à une rampe de stimulation; B) La courbe F-I présente une discontinuité, le neurone ne réagira pas à tous les courants de stimulation; C) Le neurone ne *spike* qu'une seule fois.

[PRES 2014] décrit dans le détail les particularités de ces classes et la manière dont elles peuvent être modélisées et représentées à partir d'analyses de bifurcations. Ce type d'outil mathématique sera étudié et fera l'objet d'une partie dans le chapitre 2.

1.6.2 Les limites du modèle de HH et l'importance des compartiments

Les modèles à simple compartiment sont les plus utilisés, comme suggéré précédemment, car ils sont moins coûteux en ressources et relativement efficaces, dépendant de l'application souhaitée. Néanmoins, ces modèles, malgré leurs taux de prédictions de potentiels d'action élevé où, dans le cas de Hodgkin-Huxley, de reproduction des comportements biologiques, restent limités. En effet, plusieurs études suggèrent que la morphologie du neurone possède une importance sur les *timings*, fréquences et *patterns* de trains de *spikes*. De plus, [BRETa 2015] montre qu'il y a certains phénomènes tels que l'atténuation de la tension de membrane en fonction de la fréquence ou la présence de larges variations en tension pouvant être induite par la présence de conductances actives distribuées le long de l'axone et des dendrites. Enfin, les modèles à simple compartiment considèrent que le PA est initié dans le neurone et que ce dernier est isopotentiel, c'est-à-dire qu'il possède le même potentiel en tout point de sa représentation spatiale. Or, les PA se dispersent et s'atténuent à travers les structures dendritiques et axonales. Aussi, ils sont initiés dans l'*Axone Initial Segment* (AIS) [DEBA 2011] et peuvent être initiés dans les dendrites [GASPA 2004].

[BRETa 2015] affirme que les modèles à simple compartiment sont limités. Il justifie son propos par trois arguments, trois phénomènes qui ne seraient pas représentés dans le modèle de HH. Tout d'abord, l'initiation du *spike* dans l'axone, et plus précisément le mécanisme d'initiation, aurait une importance sur la forme du potentiel d'action. Ce mécanisme particulier aurait été modélisé et étudié expérimentalement par [NAUN 2006]. Ensuite, la morphologie du neurone n'est pas représentée sur les modèles à simple compartiment. L'importance de la morphologie est mentionnée à plusieurs reprises au travers de ce manuscrit. Ces deux premiers arguments concernent des mécanismes qui influent sur la

forme et le *timing* du potentiel d'action. [DAVI 2000] propose un modèle à compartiments multiples dans lequel il montre l'importance de compartiments considérant la morphologie afin d'avoir un potentiel d'action plus proche du réel. Enfin, comme mentionné précédemment, le neurone dans le modèle de Hodgkin-Huxley est considéré comme isopotentiel. C'est-à-dire que le potentiel de la membrane est le même partout, or, un neurone, comme sa structure dendritique ou son axone, possèdera des variations en fonction de la région mesurée.

Un autre phénomène non représenté par le modèle de HH est l'apparition d'activités spontanées. La variabilité des neurones due aux fluctuations a déjà fait l'objet d'une explication dans la partie traitant de la transmission d'informations dans le cerveau. Or, ce sont des phénomènes qui sont présents et importants dans les neurones. Á noter que ces fluctuations n'apparaissent pas dans les neurones moteurs. Ce phénomène fera l'objet d'une étude sur la modélisation sous forme de bruit synaptique et d'implémentation en complément du modèle de Hodgkin-Huxley.

Les dendrites ont fait l'objet de nombreux questionnements sur leur utilité dans la communauté des neuroscientifiques. [BMEL 2006] suggère que les dendrites existent pour augmenter la surface des neurones et qu'ils peuvent étendre de 10 à 20 fois la surface du soma ce qui augmenterait le nombre possible de connexions synaptiques. [CHIKL 2002] déduisent de leurs expériences que les connexions des circuits dans la région corticale sont optimaux et que les proportions sont tous plus ou moins les mêmes. D'autres expériences appuient l'hypothèse que les dendrites existent pour pouvoir avoir plusieurs entrées possibles et séparées sur la surface des cellules nerveuses [MCBA 2001]. Les dendrites permettent également une plus grande diversité de classes de terminaux présynaptiques afin d'être sujettes à différentes lois d'apprentissages [FROE 2005] ou d'être influencées par différentes classes d'interneurones [MCBA 2001]. Outre l'aspect biochimique et biologique, dans un modèle à plusieurs compartiments, les dendrites permettent de prendre aussi en compte l'aspect morphologique. [MAIN 1996] décrit l'importance de la morphologie dendritique sur les patterns de potentiels d'action générés par les neurones. Les dendrites peuvent également initier des *spikes*. Ce phénomène est décrit et expliqué par [GASPA 2004] appuyant l'importance de la structure dendritique des neurones.

Le modèle de Hodgkin-Huxley offre la possibilité de simuler un certain nombre de phénomènes biologiques en ignorant la structure morphologique des neurones, cependant, plusieurs fonctions, comme citées précédemment, manquent et en font un modèle limité. Ainsi, un modèle à plusieurs compartiments conviendrait mieux à l'étude des maladies neurologiques. De plus, des modèles de bruits et de synapses sont nécessaires pour respectivement représenter les activités spontanées et la transmission synaptique. Des chercheurs comme [BOWE 1994] sont convaincus que la modélisation du comportement des neurones et leurs simulations avec le plus de détails possibles pourraient permettre de découvrir certains principes et mécanismes. Cependant, la complexité de ces modèles ne permet pas de créer de larges réseaux de neurones, ce qui les limite à l'étude de neurones isolés ou de comportements de petits réseaux de neurones.

1.6.3 Les niveaux de détail

Il est important de définir jusqu'où il faut simplifier un modèle pour éviter de perdre des données ou de négliger des phénomènes. Cependant, le contraire est aussi vrai, il est important de savoir jusqu'où il est possible et nécessaire de détailler le phénomène étudié. Dans le cas des neurones, les structures complexes créées par les dendrites peuvent aboutir à des modèles possédant des coûts de calculs énormes. Il en va de même dans le cas où il faut rajouter les cellules gliales, prendre en compte la myéline, les synapses, les lois d'apprentissage incluant les différents récepteurs et neurotransmetteurs, ou ajouter des types de canaux ioniques différents. Les modèles à plusieurs compartiments permettent de décrire la biochimie au niveau cellulaire et d'inclure la biophysique des synapses, des dendrites et des phénomènes retrouvés dans les axones et dans les dendrites actives [MAIN 1998]. Certains peuvent aller du modèle très détaillé et complexe comprenant 1 000 compartiments [HUYS 2006] à un modèle plus simplifié s'axant sur quatre compartiments [DAVI 2000].

Il a été montré que les neurones possèdent un système de classification complexe montrant différentes dynamiques de trains de *spikes*, différentes classes de neurones et des morphologies plus ou moins complexes de dendrites. Les modèles de neurones classiques se retrouvent limités dans le cas où il faudrait simuler des comportements bio-réaliste et surtout dans le cadre de maladies neurologiques. Plusieurs phénomènes ont été décrits et expliqués laissant ainsi une ouverture vers un type de modèles plus complexes que sont les modèles à compartiments.

1.7 La modélisation biodétaillée

Cette partie a pour but d'introduire des modèles plus détaillés en expliquant d'abord le modèle choisi, puis, les modèles à compartiments seront décrits et enfin, cette partie finira sur les modèles de synapses et de bruits synaptiques pour pouvoir simuler la présence d'activités spontanées.

1.7.1 Choix du modèle

Dans le cadre de l'étude des troubles neurologiques, nous nous limiterons aux paramètres qui ont été cités dans la partie traitant des maladies. Ainsi, nous nous sommes fondé sur un modèle « *multicompartmental* » développé par [HAY 2011] comprenant les dendrites basales, apicales et l'axone. La justification vis-à-vis de l'utilisation de ce modèle est, d'une part, que, Felix Schürmann, Henry Markram et Idan Segev font partie des créateurs de ce modèle. Ce n'est pas un argument d'autorité que de citer leurs noms, leur modèle est utilisé dans la compréhension des mécanismes des neurones mais aussi dans l'étude des maladies neurologiques qui font l'objet de publications récentes et de constantes mises à jour [MAKI 2018]. D'autre part, ce modèle possède un certain nombre de canaux ioniques et possède une structure dendritique complexe reproduisant certains des phénomènes cités dans la section précédente.

1.7.2 Modèle à compartiments multiples

Les modèles à plusieurs compartiments considèrent que les ramifications dendritiques sont plusieurs associations capacités/conductances variables reliées entre elles. La cellule

est divisée en plusieurs « compartiments » qui peuvent être le soma, l'axone et les ramifications dendritiques (voir figure 1.11).

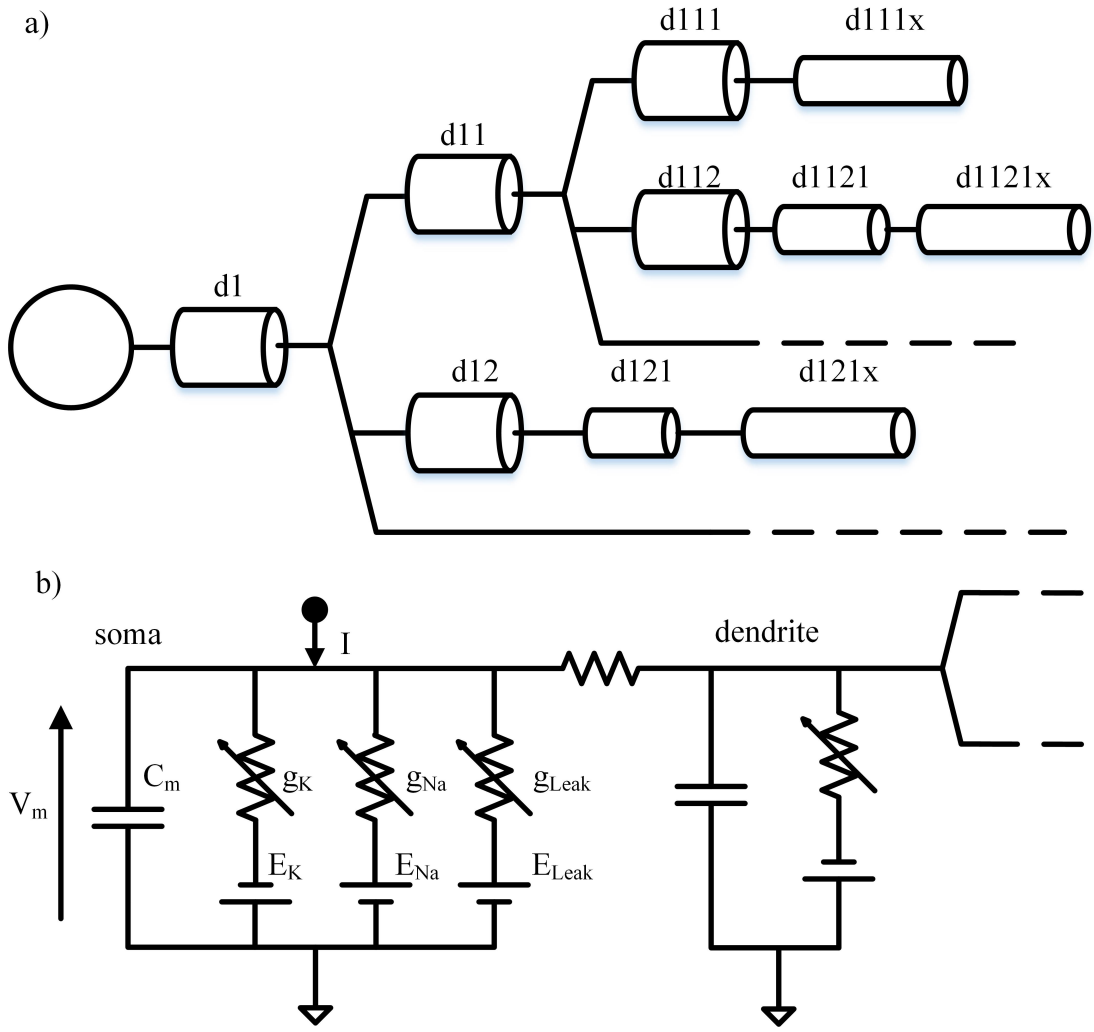

FIGURE 1.11 – a) Représentation des dendrites dans un modèle à plusieurs compartiments ; b) Schéma électrique équivalant représentant le soma connecté à une dendrite passive.

[RALL 1989] décrit sa théorie sur la résolution de la propagation de signaux dans les dendrites. Il compare les dendrites à des câbles et en déduit l'équation 1.10 représentant la relation entre le courant et la tension dans un câble à une dimension.

$$\frac{\delta V}{\delta t} + I(V,t) = \frac{\delta^2 V}{\delta x^2} \tag{1.10}$$

où, V est la tension de membrane, I le courant afférent dans la membrane, t le temps et x la distance sur le câble. L'équation 1.11 représente la discrétisation de l'équation 1.10.

$$C_j \frac{dV_j}{dt} + I_i on = \sum_k \frac{V_k - V_j}{r_j k} \tag{1.11}$$

Avec C_j la capacité de membrane du compartiment j, V_j, la tension de membrane du compartiment j, V_k les tensions de membrane des compartiments k connectés au compartiment j et $r_j k$, la résistance de la ligne de transmission séparant chaque compartiment.

On déduit de cette équation que le modèle à plusieurs compartiments intègre, pour chaque compartiment, une somme de courants provenant des compartiments adjacents.

Jusqu'ici, seul les modèles de neurones et la représentation des potentiels d'action ont été mentionnés. Cependant, comme les premières parties de ce chapitre le suggèrent, le neurone transmet ses influx nerveux via son axone qui s'interface avec les dendrites à travers les synapses. Ce dernier fera l'objet d'un état de l'art dans la partie suivante.

1.7.3 Modèle de synapses

Un modèle correspondant aux besoins de bioréalisme et biodétail a été retenu. Il s'agit du modèle présenté par [DEST 1998]. Ce modèle inclut plusieurs types de synapses à savoir les types AMPA, NMDA, $GABA_A$ et $GABA_B$. Ces noms sont issus des récepteurs de neurotransmetteurs présents sur la dendrite lors de la réception des neurotransmetteurs libérés par les vésicules. Biologiquement, on ne parle pas de neurones de types AMPA, NMDA ou GABA. Les synapses AMPA et NMDA ont une propriété excitatrice tandis que les synapses $GABA_A$ et $GABA_B$ sont inhibitrices. Les courants post-synaptiques induits par ces récepteurs sont représentés sur la figure 1.12.

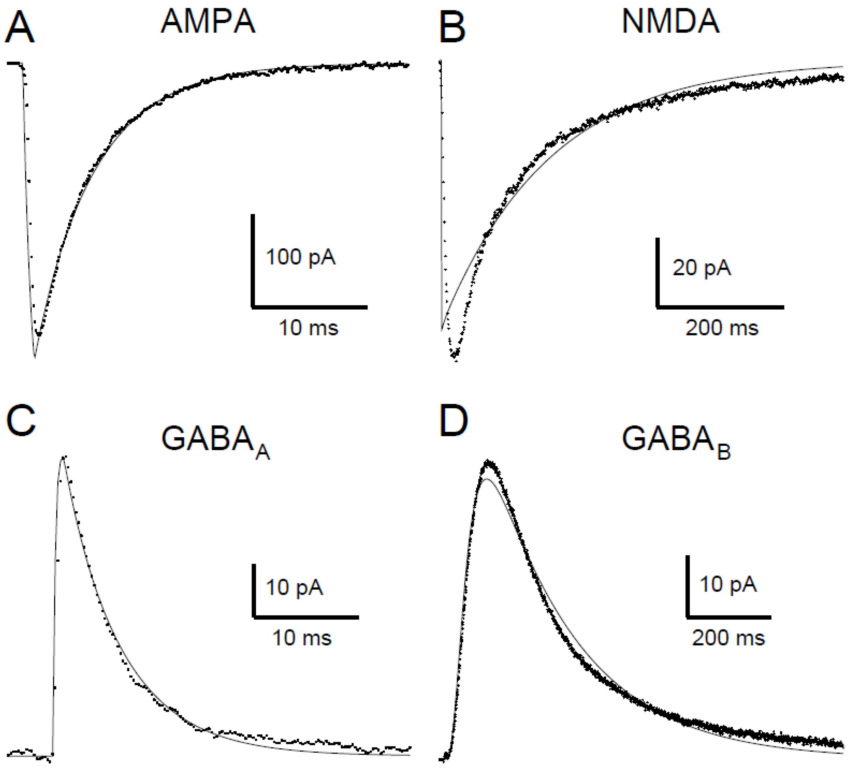

FIGURE 1.12 – Courants post-synaptiques dont les courbes ont été empruntées à [DEST 1998]. Les courbes montrent une similarité entre le modèle de [DEST 1998] et la biologie : A) Courant AMPA ; B) Courant NMDA ; C) Courant $GABA_A$; D) Courant $GABA_B$.

1.7.4 Modèles de bruit

Les activités spontanées peuvent être reproduites par l'injection d'un bruit dans le réseau de neurones généralement modélisé par une gaussienne. Cependant, il est aussi possible de générer un bruit proche de la biologie par la fluctuation des conductances des canaux ioniques [DEST 2001]. D'autres modélisent le bruit par l'ajout d'un courant bruité dans le modèle de neurone [LEVI 1968], [TUCK 1988]. Une implémentation numérique a été produite, lors de précédents travaux menés par l'équipe, en utilisant le processus d'Ornstein-Uhlenbeck. La description mathématique est représentée par l'équation 4.2 et la discrétisation pour l'implémentation numérique par l'équation 1.13.

$$dI(t) = \theta\Delta_t(\mu - I(t)) + \sigma dW(t) \tag{1.12}$$

où I est le courant bruité, θ et σ sont des paramètres permettant de « contrôler » le bruit, μ, la valeur moyenne et W le processus de Wiener.

$$I_{n+1} = I_n + \theta(\mu - I_n)\Delta_t + \sigma\Delta W_n \tag{1.13}$$

où n représente le pas de calcul.

1.8 Résumé

Plusieurs notions ont été abordées dans ce premier chapitre. Tout d'abord, nous nous sommes intéressé à la morphologie et aux mécanismes du fonctionnement des neurones. Cette description simplifiée permet de poser les bases afin de comprendre comment les neurones interagissent en utilisant des neurotransmetteurs et des influx nerveux. Plusieurs concepts ont vu le jour afin de permettre d'interpréter ces influx nerveux, appelés potentiels d'action, et les informations qu'ils pourraient transporter. Avec ces informations il est possible de revenir au sujet qui concerne les maladies neurologiques. Nous avons établi une liste de maladies comprenant les effets sur les humains et les effets sur les cellules, ce qui nous a permis d'identifier les paramètres et informations nécessaires à l'étude au niveau cellulaire. Parmi les solutions possibles du point de vue technologique pour étudier les maladies neurodégénératives, il y a la simulation des réseaux de neurones ou les interfaces cerveau/machine (BMI pour *Brain Machine Interface* en anglais). Plusieurs chercheurs ont permis la création de systèmes permettant de mesurer les activités des cellules nerveuses, de les interpréter, et de renvoyer des informations aux cellules biologiques. Certains systèmes permettraient même d'améliorer les conditions de vie des patients atteints de troubles neurologiques. La suite du chapitre s'est axée sur la modélisation et la simulation des cellules nerveuses en faisant la distinction entre les modèles bioinspirés et bioréalistes. Les modèles bioinspirés n'ont pas pour but de reproduire les activités des cellules biologiques mais d'effectuer des calculs par exemple pour la reconnaissance d'image ou le traitement d'informations. A *contrario*, les modèles bioréalistes approchent le fonctionnement des activités électriques des cellules nerveuses. Dans le cas des modèles bioréalistes, une autre distinction est faite entre les modèles à compartiment unique et les modèles à multiples compartiments. Le premier, contrairement au deuxième, ignore la morphologie du neurone et, de ce fait, ignore ou simplifie certains phénomènes présents dans le système nerveux.

Chapitre 2

Implémentation matérielle de réseaux de neurones

2.1 Introduction

Dans le premier chapitre, nous avons présenté les différents modèles choisis, cependant, ils sont complexes et requièrent beaucoup de ressources matérielles et de temps de calcul. Ce deuxième chapitre propose des solutions techniques afin de réduire les temps de calculs et d'optimiser l'implémentation. Ainsi, nous commencerons par décrire le contexte technologique, à savoir, l'état de l'art des simulations de réseaux de neurones et des méthodes utilisées dans la littérature, puis nous expliquerons celles proposées pour simplifier les équations ainsi que l'implémentation. Ce chapitre continuera sur la constitution de réseaux de neurones en implémentant les synapses et le bruit synaptique. Enfin, plusieurs méthodes de validation des méthodes d'optimisation seront présentées. Ce chapitre se terminera par les résultats des implémentations.

2.2 Contexte Technologique

Afin de respecter les besoins de bio-mimétisme et permettre les expériences biohybrides, le système doit être embarqué et temps réel. Un système temps réel prend en compte les contraintes temporelles du système étudié, mesuré ou simulé. Dans le cas des réseaux de neurones, le système devra reproduire les influx nerveux en conservant le temps biologique. Le commerce présente plusieurs choix technologiques pour ce type d'implémentation. Le développement de puces analogiques permet de concevoir des puces optimales avec une consommation très basse (nW). Cependant, le temps de réalisation est long, coûteux et non flexible. Le choix de se porter vers des systèmes numériques est donc plus judicieux pour nos applications. Le choix des supercalculateurs pose un problème d'accessibilité et de consommation. De plus, les *Central Processing Unit* (CPU) n'ont pas montré de résultats probants pour du calcul temps réel. Une autre technologie pourrait être utilisée, il s'agit des *Graphical Processing Unit* (GPU). Cependant, les machines à base de GPU ne sont pas embarquées. La technologie la plus appropriée est le FPGA, dont la description peut être retrouvée en annexe 1, qui allie flexibilité, vitesse et bas coûts. Ces systèmes sont embarqués et permettent le calcul temps réel.

2.2.1 État de l'art des implémentations de modèles bioréalistes

Les modèles de réseaux de neurones à spike peuvent être simulés sur logiciel ou implémentés sur circuits intégrés. Des supercalculateurs ont pu simuler de larges réseaux de neurones. Parmi eux, JUQUEEN [JORD 2018] permet la simulation de 18000 neurones possédants chacun 11250 synapses sur 1 seconde de temps biologique en 5 min de temps réel et dont la consommation s'élève entre 60 et 70 kW par racks sachant qu'il utilise 28 racks. K computer, un supercalculateur Japonais, met 40 minutes pour simuler 1,74 milliard de cellules nerveuses et 10,4 billion de synapses en une seconde de temps biologiques. La consommation de ces machines reste grandes et leur accessibilité réduite. Á l'opposé, les implémentations matérielles peuvent réaliser des simulations temps réels avec une faible consommation. Dans le cadre d'expériences biohybrides, le choix d'un système embarqué est donc plus pertinent. Les implémentations matérielles de réseaux de neurones peuvent être divisées en deux catégories. D'une part, les implémentations mixtes basées sur des conceptions de circuits intégrés. D'autre part, les implémentations digitales basées sur FPGA, microprocesseurs, microcontrôleurs ou *neurochips*.
Les cas d'implémentations analogiques de modèles de neurones sont nombreux. [HASL

2007] et [GEOR 2013] implémentent un modèle à compartiments incluant des dendrites reconfigurables. [SORE 2004], [BINC 2006], [RENA 2007], [LEVI 2018] et [NATA 2018] présentent leurs travaux sur des puces intégrant des modèles à conductance. [LIU 2004], [VOGEL 2004], [INDI 2007], [SCHE 2007], [QIAO 2015] et [KOHN 2016] ont utilisé des modèles à seuil pour le design de leurs puces électroniques. La plupart de ces systèmes sont composés d'un cœur analogiques qui permet de simuler le neurone. Les synapses et la plasticité sont généralement des circuits numériques.

Concernant les implémentations numériques, la majorité est basée sur des modèles bioinspirés dans le but de faire du calcul. Le nombre d'implémentation sur FPGA a augmenté ces vingt dernières années. Aussi, le niveau de détail des neurones à implémenter limite le nombre de cellules pouvant être intégrées. En effet, plus le calcul du neurone est complexe et moins il est possible d'en calculer. [NAZA 2015] présente les travaux de [CASS 2011] pour une implémentation d'un million de neurones. [ARTH 2012] montre des résultats chiffrés à 256 neurones IF pour 1 024 x 256 synapses et [WANG 2013] 4 000 neurones et 1,15 million de synapses.

Truenorth est une puce électronique créée par IBM en 2014. Elle contient un million de neurones et 256 million de synapses et permet de les utiliser en temps réel [MERO 2014]. Elle intègre des neurones en utilisant le modèle LIF (*Leaky Integrate and Fire*) et est utilisé principalement pour de la reconnaissance d'objet et de la classification. Un projet européen a été lancé en 2013, il porte le nom de *Human Brain Project* (HBP) et a pour but de faire évoluer les connaissances actuelles des neurosciences. Ce projet incluant la collaboration de plusieurs universités et laboratoires en Europe propose d'étudier le cerveau en rassemblant plusieurs domaines pour la récupération de données expérimentales, la modélisation et la simulation. Pour cette dernière, deux plateformes sont utilisées dans le projet : SpiNNaker et BrainScaleS. La plateforme analogique BrainScales peut aller jusqu'à 4 million de neurones avec 880 millions de synapses. La plateforme numérique SpiNNaker possède, quant à elle, une architecture à plusieurs processeurs permettant de simuler jusqu'à 1 million de neurones. Les neurones utilisés sont des LIF [ALBA 2018].

Enfin, parmi les grands projets du moment, Intel propose sa puce Loihi, la dernière d'une série de puces neuromorphiques, contenant 130 000 neurones à *spikes* et 130 millions de synapses [DAVI 2018].

2.2.2 Les implémentations FPGA

La complexité des circuits analogiques et la consommation des supercalculateurs ont amenés à remettre en cause l'utilisation de ces technologies pour la simulation de réseaux de neurones. L'utilisation de circuits numériques fait l'objet d'une large littérature dans l'implémentation de réseaux de neurones. Nous nous intéresserons ici aux implémentations de HH et des modèles à plusieurs compartiments.

Hodgkin-Huxley

Le modèle de Hodgkin-Huxley, du fait de sa complexité, a été le sujet de peu d'études concernant une implémentation sur FPGA. Selon la méthodologie utilisée et les techniques numériques, le temps de calcul, le nombre de neurones et la précision peuvent être diffé-

rents. [OSOR 2016] présente une architecture pipeline utilisant des calculs à base de nombres flottants et des méthodes complexes telles que les algorithmes de Runge-Kutta et de Goldsmith comme méthodes de résolutions et des développements en série de Taylor pour les opérateurs complexes. Ces méthodes requièrent beaucoup de temps et de ressources. [BONA 2014] montre de bonnes précisions sur les calculs en utilisant des algorithmes rapides et peu coûteux en ressources comme le *COordinate Rotation DIgital Computer* (CORDIC) et la méthode d'Euler, ce qui leur permet de simuler 120 neurones connectés par des synapses. Plus récemment, [AKBA 2018] a présenté une autre manière de calculer des neurones en utilisant le modèle de HH permettant d'augmenter le nombre de neurones implémentés à 5120 en temps réel. Ces travaux montrent l'évolution des techniques numériques d'implémentation. En revanche, les implémentations présentées ne contiennent que des neurones de type FS. La morphologie n'est pas prise en compte, aussi, aucun ne présente un moyen rapide et efficace de rendre le système flexible, c'est-à-dire de modifier les paramètres biophysiques en temps réel.

Simplification

La suite de ce chapitre traitera des méthodes de simplification mis en place pour réduire les ressources (LUT, Flip-Flop et DSP) prises sur le FPGA. Elle se composera en trois parties : 1) L'état de l'art des simplifications faites sur les modèles de Hodgkin-Huxley ; 2) Les simplifications mathématiques adaptées au numérique ; 3) La simplification électronique concernant l'aspect plus technique de l'implémentation FPGA. Dans un premier temps, la simplification sera appliquée au modèle de Hodgkin-Huxley utilisant les paramètres de [POSP 2008]. Ce modèle reproduit les activités de neurones corticaux et notamment les neurones de type FS, RS, IB et LTS. Dans un second temps, une adaptation des modèles de synapses et de bruits sera montrée. Des résultats seront donc présentés sur un réseau de neurones de Hodgkin-Huxley complet. Plusieurs méthodes de validation permettront de montrer le bon fonctionnement temps-réel et le bioréalisme du système. Enfin, ce manuscrit exposera une proposition de méthode d'implémentation d'un modèle à plusieurs compartiments de Hodgkin-Huxley.

2.2.3 Méthode de simplification

De nombreuses méthodes de simplification des équations de Hodgkin-Huxley pour une implémentation sur FPGA sont décrites. Certaines méthodes proposent des approximations du modèle de Hodgkin-Huxley afin de faciliter les calculs tout en gardant le bio-réalisme du modèle. [IZHI 2003], [BRET 2005] et [KOHN 2016] décrivent des équations à deux variables permettant de reproduire la simulation de réseaux de neurones. Cependant, il est important de définir jusqu'où la simplification peut aller et quels sont les critères permettant de choisir au mieux la méthode permettant de réduire les équations. Le chapitre 1 a mis l'accent sur l'importance de conserver les paramètres biophysiques des équations. De ce fait, les modèles issus de la simplification de Hodgkin-Huxley tels que Morris Lecar, Izhikevich ou Fitzgumoh Nagumo ne seront pas retenus. Les méthodes permettant d'optimiser le modèle de HH à une implémentation matérielle peuvent prendre plusieurs formes. D'une part, il est possible de modifier les équations mathématiques afin de les adapter au FPGA. D'autre part, des techniques, des algorithmes et des méthodes de conception peuvent permettre une implémentation plus optimale. L'état de l'art concernant les implémentations numériques de modèles bioréalistes décrit dans le premier chapitre propose des méthodes de simplifica-

tion numérique. Il y a peu de modifications des équations de Hodgkin-Huxley mais le plus souvent une proposition de méthodes de calcul numérique en appliquant des méthodes de régressions et d'interpolation. Ces cas seront énoncés et développés dans le chapitre 3 de ce manuscrit. Ces méthodes peuvent impacter le temps de calcul et le nombre de ressources. [AKBAR 2018] propose une méthode basée sur une interpolation impliquant le calcul de 2048 neurones connectés en temps réel en utilisant un nombre limité de ressources. Cependant, l'implémentation proposée simule le modèle à une fréquence de 58,8 MHz ce qui constitue une fréquence relativement faible et montre les limitations du système. De plus, seul un type de neurone est représenté : le neurone FS.

2.2.4 Le cas du FPGA choisi

Le FPGA que l'équipe possède est un Kintex 7 XC7K325T-2FFG900C de la marque Xilinx implémenté dans une carte électronique vendue sur le marché et appelé Genesys 2. La documentation indique que chaque *slice* contient 4 LUT avec 6 entrées (potentiellement 26 bits de mémoire), 8 DFF, une ALU et des MUX et chaque CLB contiendrait 2 *slices*. Ce FPGA possède 50 940 slices soit 203 800 LUT, environ 16 Mbits de RAM, 840 DSP, une horloge de 200 MHz et d'autres composants que nous n'avons pas estimé pertinent pour l'implémentation des réseaux de neurones.

2.2.5 Paramètres de [POSP 2008]

[POSP 2008] intègre dans le modèle de Hodgkin-Huxley trois types de canaux ioniques supplémentaires. Le circuit équivalent basé sur [HODG 1952] devient donc celui de la Figure 1.

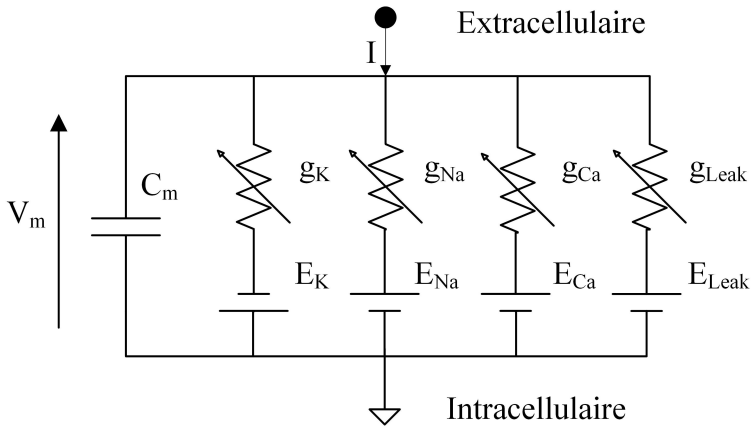

FIGURE 2.1 – Circuit équivalent à la membrane des neurones corticaux utilisant les paramètres de [POSP 2008]. Un courant slow-potassium et deux courants calcium, un pour les *burst* du type IB et un autre pour les rebonds du type LTS sont ajoutés par rapport au circuit équivalent de Hodgkin-Huxley (Cf. Chapitre 1).

La figure 2.1 permet d'obtenir l'équation 2.1.

$$C_m \frac{dV_m}{dt} = I_s - \sum_i I_i \tag{2.1}$$

	I_K	$I_N a$	I_M	$I_C aL$
p	4	3	1	2
q	0	1	0	1

TABLE 2.1 – paramètre p et q qui servent respectivement de puissance pour les variables d'activation et d'inactivation.

	FS	RS	IB	LTS
dneur (µm)	67	96	96	96
Sm (cm²)	$\pi.d_{neur}{}^2.10^{-8}$			
Cm (µF/cm²)	1			
Is (µA/cm²)	3,54	2,59	0,51	1,24
(mS/cm²)	10	5	5	5
(mS/cm²)	50	50	50	50
(mS/cm²)	0,15	0,1	0,01	0,01
(mS/cm²)	0	0,07	0,03	0,03
(mS/cm²)	0	0	0,17	0
(mS/cm²)	0	0	0	0,4
(mV)	-100			
(mV)	50			
(mV)	-70	-70	-85	-50
(mV)	0	0	120	120
VInit (mV)	-70	-70	-84	-84

TABLE 2.2 – Valeurs utilisées pour le calcul des neurones en utilisant les paramètres de [POSP 2008]. dneur et Sm sont respectivement le diamètre et la surface du neurone.

où V_m est la tension de membane, Cm la capacité de membrane, I_s le courant de stimulation et I_i les courants ioniques correspondants aux ions potassium, sodium, de fuite, slow potassium et calcium, avec i ∈ K, Na, M, CaL, CaT. Les courants ioniques peuvent être représentés par les équations 2.2, 2.3 et 2.4.

$$I_i = \bar{g}_i m^p h^q (V_m - E_i) \tag{2.2}$$

$$I_C aT = \bar{g}_T {m_\infty}^2 h(V_m - E_C a) \tag{2.3}$$

$$\frac{dX}{dt} = \frac{x_\infty(V_m) - x}{\tau_x(V_m)} \tag{2.4}$$

où x ∈ m, h, $\tau_x(V_m)$ et $x_\infty(V_m)$ sont obtenus par les équations 2.5 et 2.6, p et q sont résumés dans le tableau 2.1. Les variable m et h sont respectivement les variables d'activation et d'inactivation des canaux voltage-dépendants.

$$\tau_x(V_m) = \frac{1}{\alpha_x(V_m) + \beta_x(V_m)} \tag{2.5}$$

$$x_\infty(V_m) = \frac{\alpha_x(V_m)}{\alpha_x(V_m) + \beta_x(V_m)} \tag{2.6}$$

Les paramètres originaux donnés par [POSP 2008] sont regroupés dans le tableau 2.2.

		a_1	b_1	c_1	d_1
I_K	m	2^{-1}	$2^{-5}+2^{-6}$	-1,128	0,467
I_{Na}	m		2^{-4}	-1,718	0,497
	h		-2^{-3}	2,705	0,502
IM	m		$2^{-5}+2^{-6}$	-0,938	0,5
I_L	m		2^{-3}	4,151	0,495
	h		$-2^{-6}-2^{-7}$	2,705	0,502

TABLE 2.3 – Paramètres pour le calcul des x_∞ ; les calculs de IB et de LTS requièrent plus de précision et ne peuvent donc pas être arrondis à une puissance de 2. De ce fait, les paramètres deviennent : $a_1 = 0.001179 * v + 0.5479$, b_1=2-4 + 2-10, $c_1 = 1.954$ et $d_1 = 0.4577$ lorsque $V_m < -30$ sinon, $a_1 = 0.0001474 * v + 0.4383$, $b_1 = 2-4+2-8$, $c_1 = 1.817$ et $d_1 = 0.567$.

		a_1	b_1	c_1	d_1
I_K	m	1,325	0,046	-0,497	0,335
I_{Na}	m	0,893	-0,041	0,937	0,029
	h	γ	0,099	-1,960	0,3498
I_M	m	2^{-1}	-1,618	2^{-4}	0,498
I_L	m	δ	0,093	3,452	0,371
	h	ϵ	-0,029	-1,833	150,7

TABLE 2.4 – Paramètres pour τ_x où $\gamma = -0,2334(V_m - V_x) + 9,649$, $\delta = -0,111V_m + 2,264$ et $\epsilon = -0,9446V_m + 237,6$; $V_x = 2mV$.

2.2.6 Adapter les mathématiques au numérique

Régression

Une des méthodes proposées est de faire correspondre une équation plus simple à implémenter en numérique avec les équations originelles. Plusieurs critères de régressions doivent être pris en compte afin de répondre aux besoins. Les nouvelles équations doivent bien entendu être plus efficaces que les originaux mais aussi permettre d'effectuer des calculs rapides en un nombre de cycles d'horloge réduits. Le nombre de cycles d'horloge détermine le nombre d'opérations possibles entre chaque pas de calcul. De plus, l'implémentation devra comporter le moins de ressources FPGA (LUT, bascules D, RAM). En effet, la réduction de ressources implique une réduction de la consommation et la possibilité d'ajouter, d'une part, de nouvelles fonctions au réseau et, d'autre part, d'autres neurones. Il est possible de représenter les équations précédentes par les équations 2.7, 2.8 et 2.8.

$$\frac{dx}{dt} = \frac{x_\infty(V_m) - x}{\tau_x(V_m)} \tag{2.7}$$

$$x_\infty(V_m) = a_1 \tanh(b_1(V_m - V_x) + c_1) + d_1 \tag{2.8}$$

$$\tau_x(V_m) = \frac{a_2}{\cosh(b_2(V_m - V_x) + c_2)} + d_2 \tag{2.9}$$

où x est l'ion calculé (potassium, sodium ou calcium), Vm la tension de membrane et a, b, c et d des paramètres permettant de faire correspondre les nouvelles courbes avec les courbes originales. Ces constantes sont présentes dans les tableaux 2.3 et 2.4.

	a	b	c	d
minf	2^{-1}	$2^{-4}+2^{-6}$	5	0,5
uinf		-2^{-3}	-11	
tauu	-28,09	0,162	12,808	37,03

TABLE 2.5 – Paramètres pour l'équation du courant ionique décrivant le courant IT.

Le courant I_T suit les mêmes règles à la différence que τ peut être également calculé en utilisant une tangente hyperbolique. De ce fait, m_∞, u_∞ et τ_u peuvent être calculés en suivant la forme de l'équation 2.10 et en utilisant les paramètres du tableau 2.5.

$$f(V) = a\tanh(bV_m + c) + d \tag{2.10}$$

Il a été choisi d'utiliser des tangentes hyperboliques et des cosinus hyperboliques car il existe des algorithmes et méthodes simples et rapides à implémenter sur FPGA permettant de calculer ces fonctions. Le *COordinate Rotation Digital Computer* (CORDIC) permet d'effectuer grâce à des opérations d'addition, de soustraction et de décalage de bits, des fonctions hyperboliques [VOLD 1959]. Il est aussi possible de calculer une tangente hyperbolique par interpolation [AHME 2017]. Ces méthodes feront l'objet d'une description plus complète dans le prochain chapitre destiné aux implémentations sur FPGA. De plus, en procédant ainsi, il est possible d'utiliser un seul module de calcul pour tous les courants ioniques sauf pour le courant IT dont les équations sont différentes. De plus, certains paramètres peuvent être approximés sous forme de puissance de 2 ou de somme de puissances de 2 afin de remplacer les éventuelles multiplications/divisions lors de l'implémentation sur FPGA par des opérations de décalage de bits. Les résultats de ces équations sont présents sur la figure 3 et sont comparés avec les résultats des équations originales. Les signaux de type FS (Figure 3.a) et IB (Figure 3.c) montrent une fréquence et une amplitude fortement corrélées. Les différences apparaissent sur le RS (Figure 3.b) qui montre une légère différence en fréquence et le neurone LTS (Figure 3.d) qui affiche une bouffée de *spikes* différente.

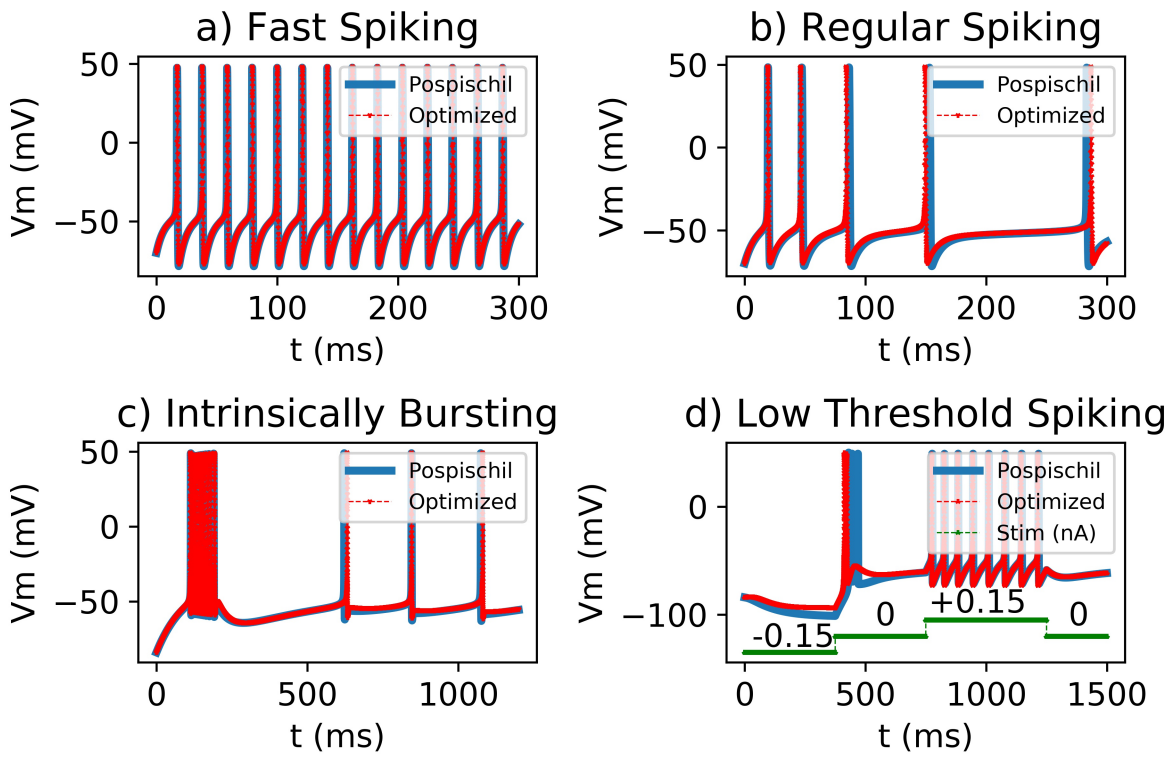

FIGURE 2.2 – Comparaison des différents types de neurones simulés via le modèle proposé par pospischil et simulés par le modèle optimisé. Les neurones FS, RS et IB sont stimulés par un courant constant; a) FS, les deux courbes sont quasi confondues; b) RS, une légère différence apparaît et impacte la fréquence de *spike*; c) IB, les *bursts* contiennent le même nombre de *spikes* mais sont de fréquence différentes. Cependant, les autres *spikes* sont quasi confondus; d) LTS, lors du passage de $-0,15$ nA de stimulation à 0, les *bursts* contiennent un nombre de *spikes* différent.

Le choix des équations et des paramètres a été fait en fonction des possibilités d'implémentation qu'offre le FPGA. Ces nouvelles équations permettent de générer des potentiels d'action similaires aux équations d'origine. Ces méthodes peuvent aussi bien s'appliquer au calcul de neurones qu'au calcul des synapses qui feront l'objet de la prochaine section.

2.3 Réseau de neurones

2.3.1 Synapses

Le modèle de synapses choisi est celui de [DEST 1998] car, comme mentionné dans le premier Chapitre, il s'agit d'un modèle proche de la biologie, comme le modèle de neurone, et comporte plusieurs récepteurs de neurotransmetteurs. Il prend en compte les récepteurs de type AMPA, NMDA, GABAa et GABAb. Il est modélisé, comme la plupart des modèles de synapses, par des courants synaptiques qui s'additionneront aux autres courants ioniques dans le modèle de Hodgkin-Huxley. Les équations 2.11 et 2.12 représentent le fonctionnement des synapses excitatrices (AMPA et NMDA) et les équations 2.13 et 2.14 les synapses inhibitrices (GABAa et GABAb).

	AMPA	NMDA	GABAa	GABAb
α	1,1	0,072	5	
β	0,19	0,00066	0,18	
g (nS)	0.9	0.35	1	1
E (mV)	0	0	-70	-95
K1				0,09
K2				0,0012
K3				0,18
K4				0,034
Kd				100
n				4

TABLE 2.6 – Récapitulatif des paramètres utilisés dans [DEST 1998] pour les équations des courants synaptiques.

$$I_{AMPA} = \bar{g}_{AMPA} r (V_{post} - E_{AMPA}) \tag{2.11}$$

$$I_{NMDA} = \bar{g}_{NMDA} r B(V_p re)(V_{post} - E_{NMDA}) \tag{2.12}$$

$$I_{GABAa} = \bar{g}_{GABAa} r (V_{post} - E_{GABAa}) \tag{2.13}$$

$$I_{GABAb} = \bar{g}_{GABAb} \frac{s^n}{s^n + K_d} (V_{post} - E_{GABAb}) \tag{2.14}$$

où,

$$\frac{dr}{dt} = a[T(V_{pre}](1-r) - \beta r \tag{2.15}$$

$$\frac{ds}{dt} = K_3 r - K_4 s \tag{2.16}$$

$$\frac{dr}{dt} = K_1[T(V_p re)](1-r) - K_2 r \tag{2.17}$$

I_x est le courant synaptique avec x ∈ { AMPA, NMDA, GABAa, GABAb }, Vpre la tension de membrane du neurone émetteur et V_{post} du neurone récepteur. α, β, K_d, K_1, K_2, K_3, K_4 et n sont des paramètres constants et g_x est la conductance des récepteurs représentés par x. Les équations et les paramètres peuvent être trouvés dans [DEST 1998] et sont résumés dans le tableau 2.6.

Les récepteurs représentent différents comportements qui peuvent être trouvés dans le système nerveux. AMPA et NMDA induisent respectivement une rapide et une lente excitation au neurone. GABAa et GABAb sont des synapses inhibitrices qui produisent respectivement une rapide et lente inhibition.

Les dendrites ont une fonction intégratrice à la réception des tensions de membranes présynaptiques. Ainsi, le PSP (*Post-Synaptic Potential*) dépend de la somme des courants synaptiques. La tension de membrane d'un neurone *n* peut donc être décrite par les équations 2.18 et 2.19 :

$$\frac{dV_j}{dt} = I_s - \sum I_{ion_k} + \sum w_{ij} g_{ij} m_j(V_j)(V_j - E_{ij}) \tag{2.18}$$

$$\frac{dV_j}{dt} = I_s - \sum I_{ion_k} + m_j(V_j) \sum w_{ij} g_{ij}(V_j - E_{ij}) \tag{2.19}$$

avec i l'index du neurone considéré, j l'index de la synapse considérée, w_{ij}, g_{ij} et E_{ij} respectivement le poids synaptique, la conductance et le potentiel d'équilibre de la synapse conduisant le potentiel du neurone i vers le neurone j, m la fonction d'activation et d'inactivation des récepteurs, I_{ion_k}, les courants ioniques avec k correspondant aux ions sodium, potassium, calcium et fuite et Is le courant de stimulation. L'opération devient donc un calcul matriciel où w, g et E sont des matrices et m et V sont des vecteurs. Le poids synaptique w et la conductance g peuvent être sauvegardés dans la même matrice en multipliant les deux termes. Dans un cas simple, w est une matrice dont les termes seront égaux soit à 1 (présence d'une synapse) soit à 0 (absence de synapse).

2.3.2 Bruit

Le modèle de bruit a été présenté dans le Chapitre 1. Cette section portera donc sur son implémentation. Le module VHDL permettant de générer un courant stochastique a été réadapté de [GRAS 2016]. La réadaptation inclut une version « pipeline » du module original permettant de générer un bruit différent pour chaque neurone. L'équation 2.20 représente la discrétisation du bruit d'Ornstein-Ulhenbeck.

$$I_{n+1} = I_n + \theta(\mu - I_n)\Delta_t + \sigma \Delta W_n \tag{2.20}$$

avec, I est le courant bruité, θ et σ sont des paramètres permettant de « contrôler » le bruit, μ, la valeur moyenne et W le processus de Wiener. Le processus de Wiener peut être reproduit grâce à un registre à décalage à rétroaction linéaire, soit en anglais, *Linear Feedback Shift Register* (LFSR). Ce type de circuit permet de réaliser de manière peu coûteuse des générateurs de bits pseudo-aléatoires. L'implémentation de ce type d'équation peut être validée en comparant les distributions, d'une part, du processus de Wiener et, d'autre part, le courant simulé sur ordinateur (MATLAB) et sur FPGA. Il existe une méthode statistique permettant de représenter graphiquement la proximité entre deux distributions. La figure 2.3.a représente les quantiles de la fonction pseudo-aléatoire sur FPGA en fonction des quantiles de la même fonction sur MATLAB. Plus le nuage de points bleus est proche de la droite rouge et plus les distributions sont similaires. La figure 2.3.b montre que le courant bruité simulé sur ordinateur et généré sur FPGA est similaire.

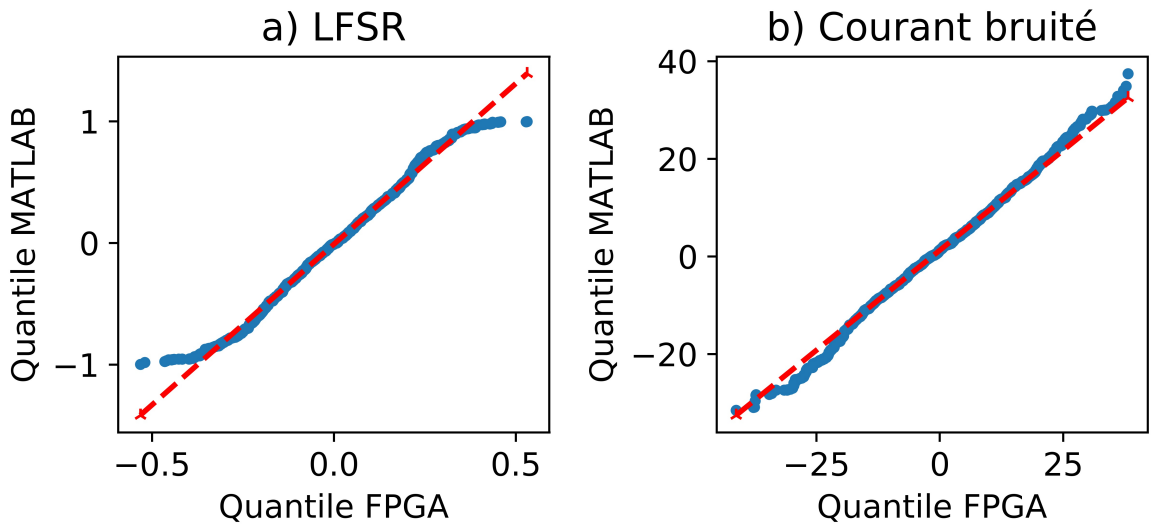

FIGURE 2.3 – Graphique sous forme de Quantile-Quantile permettant de comparer deux distributions. Plus les points bleus sont proches de la droite pointillée rouge, plus les distributions comparées sont similaires ; a) Comparaison entre le LFSR implémenté sur FPGA issue des travaux de [GRASS 2016] avec la fonction « randn » de Matlab ; b) Comparaison du courant bruité utilisant les équations de OU entre l'implémentation FPGA et la simulation Matlab.

Des méthodes de régression ont été utilisées afin de décrire le fonctionnement des courants ioniques par des équations alternatives constituant une simplification optimisée pour une implémentation numérique. Ces simplifications entraînent une baisse de la précision et une différence plus ou moins légère sur le potentiel d'action. La suite de ce manuscrit aura pour objectif de quantifier ces différences et de valider des simplifications.

2.4 Validation

Cette section se découpera en trois parties. Tout d'abord, nous parlerons des méthodes utilisées pour comparer des activités de neurones entre elles. Ensuite, nous parlerons des outils permettant de comparer deux signaux. Enfin, nous parlerons de l'analyse de bifurcations.

2.4.1 Comparer des potentiels d'action

De nombreux outils ont été développés afin de quantifier les influx nerveux et de les comparer aux modèles mathématiques [COHE 2013]. D'une part, des méthodes permettent d'évaluer la dynamique des activités du/des neurone(s) en se basant sur une fréquence moyenne de *spikes* en utilisant des analyses de corrélation comme [GRESC 2011] dans le cadre de l'étude de la rétine. D'autre part, d'autres méthodes servent plutôt à évaluer le neurone au niveau du *spike* comme l'utilisation d'analyses de corrélations sur les moments où un potentiel d'action survient sur plusieurs expériences [MAIN 1995]. Parmi ces outils, il est possible de représenter les fréquences de *spikes* sous forme d'une moyenne ou d'une distribution. La moyenne des intervalles entre *spike* d'un train de potentiels d'action peut donner une information sur la variabilité du signal.

En prenant l'exemple de [SHEVE 1995] montrant la variation de la fréquence moyenne des *spikes* dans les neurones corticaux d'un chat en fonction de l'angle de l'image d'un trait, il est possible de tracer une distribution des fréquences de *spikes* en fonction de l'angle du trait projeté. Cette distribution peut aussi être représentée en fonction du temps. L'intérêt est d'observer la variabilité des neurones sur plusieurs expériences. Dans le cas de modèles, cette méthode permet de comparer la dynamique des neurones réels avec les neurones simulés. Une autre méthodologie consiste à étudier les corrélations entre le nombre de *spikes* générés par une cellule nerveuse et la simulation d'un modèle représentant ce qui est appelé un facteur de corrélation. [COHE 2013] regroupe et étudie des outils de corrélation permettant d'évaluer des signaux neuronaux. D'autres travaux tels que ceux de [SHIGE 2003] ont permis le développement d'une méthode statistique permettant d'observer l'évolution de la dynamique des neurones à travers les distributions des intervalles entre *spikes*. L'inconvénient de cette méthode est sa complexité comparée aux autres méthodes qui vont être présentées par la suite.

Il a été mentionné dans le chapitre 1, section « classification », l'utilisation de courbes F-I. Pour rappel, ce type de courbes permet de représenter graphiquement l'évolution en fréquence des influx nerveux en fonction de leur excitabilité en courant. Ce type de courbes permet par exemple de représenter les différences qu'il peut y avoir sur différentes familles de neurones ou de comparer les évolutions des neurones biologiques avec les neurones artificiels. [POSP 2011] présente des courbes F-I de neurones corticaux de rats concernant les neurones de type FS et RS et les compare aux modèles qu'il propose. Nous utiliserons ce type de courbes pour valider notre implémentation.

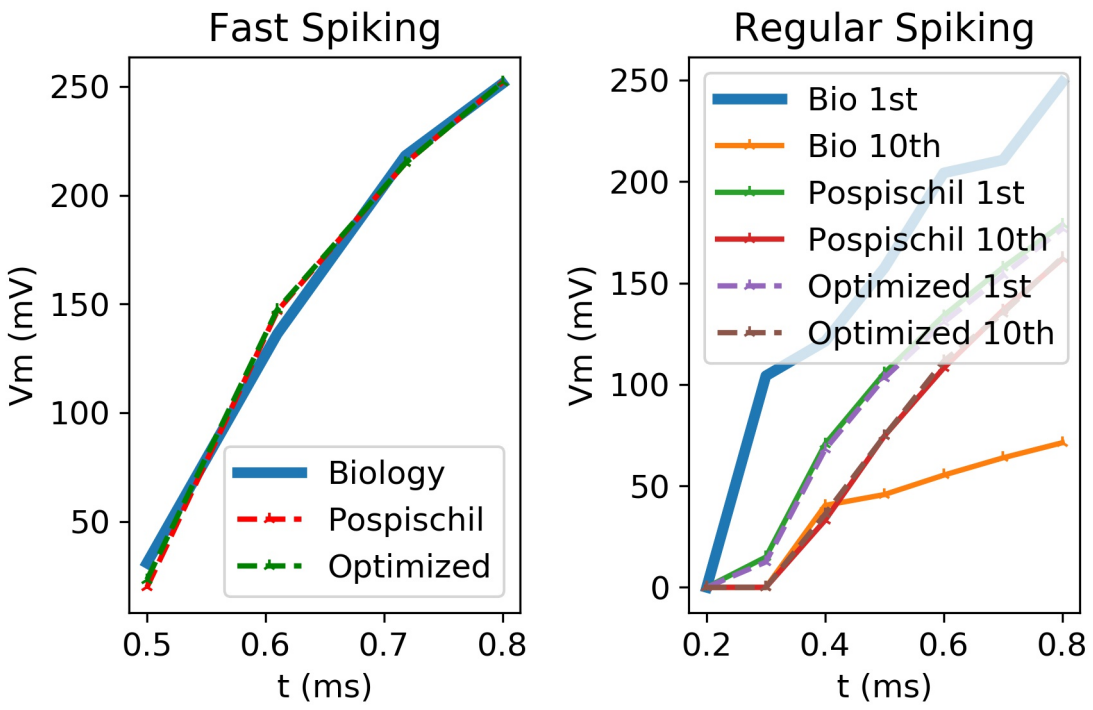

FIGURE 2.4 – Courbes F-I : a) *Fast Spiking* (FS), la tendance des deux courbes sont proches de la biologie; b) *Regular Spiking* (RS), représentant la fréquence des 1er et 10e *spikes*. Le comportement biologique est éloigné du modèle, cependant, l'optimisation apportée sur les équations affiche un comportement biologique proche du modèle présenté par [POSP 2011].

La figure 2.4 représente les courbes F-I des neurones FS et RS comparées aux résultats issus de la biologie et des modèles présentés par [POSP 2011]. Il est important de noter deux choses. D'une part, le modèle ne reproduit pas les activités biologiques des neurones de type RS. Ce problème est présenté et assumé par l'auteur qui montre tout de même qu'il y a bien une adaptation en fréquence dans le modèle qui est similaire à la biologie. D'autre part, ce type de courbes ne peut représenter les neurones de types IB et LTS. Des chercheurs de l'université de Kyoto ont publié une méthode statistique permettant de mesurer et d'analyser graphiquement la similarité des potentiels d'action des neurones FS, RS, IB et LTS [SHIGE 2003]. Cependant, comme précisé précédemment, l'outil est complexe et les méthodes de *cross-correlation* [IDE 2017] offre tout autant la possibilité d'analyser des signaux issus de neurones.

2.4.2 Quantifier la différence

Ces problématiques d'analyse et de quantification de la corrélation entre deux signaux ont fait l'objet d'études sur des domaines différents comme l'analyse de données dans l'imagerie médicale [IDE 2017]. Ces dernières relatent, dans le cadre des neurosciences, l'utilisation d'outils qui ont été développés afin d'étudier les corrélations dans les réseaux de neurones ou, comme dans le cas de [CORRE 2009], dans les analyses de signaux issus des *Imageries par Résonance Magnétique* (MRI) et EEG. Des méthodes statistiques permettent l'analyse de signaux issus du cerveau et d'imagerie médicale [IDE 2017]. Cependant, elles

restent complexes et ne sont pas pertinentes dans le cadre de l'analyse de neurones isolés ou de petits réseaux de neurones. Une autre méthode existe, le *Pearson Product Moment Correlation* (PMCC) qui permet de quantifier les différences entre signaux en calculant un réel appelé « r » compris entre −1 et 1 [COHE 2013]. L'équation 2.21 représente le calcul de la PPMC en fonction des signaux à comparer.

$$r_x y = \frac{1}{N-1} \sum_{i=1}^{N} (\frac{x_i - \mu x}{\sigma_x})(\frac{y_i - \mu_y}{\sigma_y}) \tag{2.21}$$

où μ_x et μ_x sont respectivement la moyenne de x et de y, σ_x and σ_y sont les écarts types de x et y, N est le nombre d'échantillons.

L'outil appelé *cross-correlation* est également une méthode souvent utilisée, tous domaines confondus, dans l'analyse de différences entre signaux. Il donne une valeur r qui est cette fois comprise entre 0 et 1 mais fournit aussi une information sur le décalage temporel que les deux signaux peuvent présenter l'un par rapport à l'autre. L'avantage de cette méthode est qu'elle prend en compte les potentiels déphasages. Elle est souvent comparée au produit de convolution dont le calcul est très proche.

$$r_x y(i) = \frac{1}{N-1} \sum_{i=1}^{N} x(i) y(i+j) \tag{2.22}$$

où N est le nombre d'échantillons, i et j les index des vecteurs de données.

Les avantages des deux dernières méthodes citées sont la simplicité de calcul, la représentation graphique et quantitative des différences. En effet, la PPMC peut être représentée par un nuage de points (*scatterplot* en anglais) représentant les grandeurs mesurées en fonction de la grandeur de référence. Si les signaux sont les mêmes, la courbe formée sera une droite d'équation $f(x) = x$. Ainsi, si le signal formé par un nuage de points tend vers cette droite $f(x) = x$, les signaux sont alors corrélés. Dans le cas contraire où les points seraient éloignés de la droite de référence, les grandeurs à comparer ne sont pas corrélées. La *cross-correlation*, quant à elle, peut être aussi représentée sous la forme d'un décalage (en ms dans notre cas) en fonction de cette valeur R. Ces deux méthodes ont été utilisées dans les travaux présentés dans ce manuscrit car elles possèdent un bon rapport simplicité d'utilisation et efficacité. Le PPMC dans le cas de l'analyse d'un seul *spike* et la *cross-correlation* dans le cadre d'un train de *spikes*.

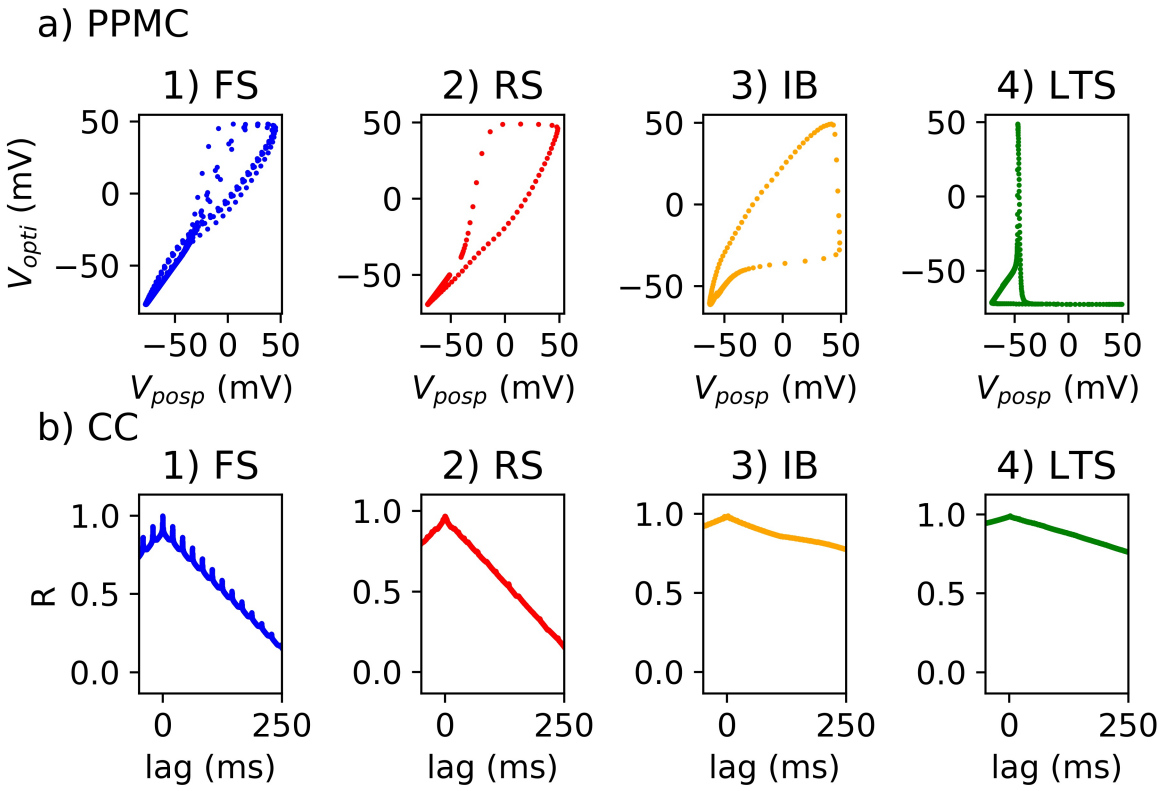

FIGURE 2.5 – a) un seul spike représenté sous forme d'un nuage de points pour les différents types de neurones ; b) représentation de la valeur R en fonction du lag dans le cas de la cross-correlation ; 1) Neurone de type FS à 0,5 nA de stimulation, $rCC = 0,99$ et $rPPMC = 0,96$; 2) Neurone RS à 0,75 nA de stimlation, $rCC = 0,97$ et $rPPMC = 0,91$; 3) Neurone de type IB avec 0,15 nA de stimulation $rPPMC = 0,89$ et $rCC = 0,97$; 4) Neurone de type LTS avec une variation de -0,15, 0 puis 0,15 nA $rPPMC = 0,34$ et $rCC = 0,97$.

2.4.3 Analyse de systèmes dynamiques

La modélisation mathématique de phénomènes de complexité variable a amené à étudier les systèmes dynamiques. Un système dynamique est un ensemble de variables décrivant l'état du système et les lois régissant l'évolution de ce système. Il est défini par un processus déterministe modélisé mathématiquement. Il représente dans divers domaines tels que la physique, la chimie, la biologie ou l'économie, un modèle permettant d'effectuer des prédictions passées, présentes et futures. Il est souvent représenté par des équations différentielles fonctions du temps dont les états futurs dépendent de paramètres et des états initiaux. Ces modèles peuvent parfois devenir complexes, ainsi, des méthodes ont été développées afin de permettre leur analyses. La méthode utilisée ici sera l'analyse de bifurcations qui permet d'observer la dynamique d'un système en fonction de ses paramètres et de ses états initiaux. On peut ainsi graphiquement représenter le comportement d'un système à base d'équations différentielles ordinaires, sa sensibilité aux paramètres donnés et sa stabilité. Cette partie n'a pas pour but de détailler ni d'expliquer toute la riche et complexe théorie de l'analyse de bifurcations. Elle se contentera de décrire brièvement les notions et de les appliquer à notre cas : le modèle de Hodgkin-Huxley. Dans ce cas d'étude, cette analyse permet implicitement de déterminer le comportement du neurone étudié. Ainsi, il est possible d'observer des similitudes entre le comportement d'un neurone biologique et le

comportement d'un modèle de neurone.

2.4.4 Stabilité/instabilité

Avant d'aller plus loin, plusieurs notions sont essentielles. Tout d'abord, on parle de plan ou portrait de phase lorsqu'on représente la dérivée d'un système en fonction de lui-même. Les analyses de bifurcations sont souvent représentées sur un plan de phase, car la dynamique d'un système est représentée par sa dérivée. Ainsi, toute variation peut être observée lorsqu'elle est représentée graphiquement. Sur ce plan de phase, il est possible de visualiser des points représentant le moment où la dérivée est nulle. Ils sont appelés « points d'équilibre » et constituent les « *nullclines* » ou « *isoclines* ». Les fonctions dont les solutions sont les points d'équilibre peuvent être représentées sous la forme d'une matrice Jacobienne. L'affichage graphique de ces fonctions donne un champ de vecteurs déterminant la direction que prendra la variation du système depuis un point d'équilibre. Lorsque les solutions peuvent converger vers un point d'équilibre ou diverger depuis ce dernier, on parle alors de bi-stabilité. Le système atteint un équilibre à ces points jusqu'à ce qu'une variation d'un paramètre entraîne l'instabilité du système. Ce cas de figure représente la dynamique d'un neurone représenté par les équations de Hodgkin-Huxley. Le système est à son état d'équilibre jusqu'à ce qu'il reçoive un courant de stimulation suffisant pour lancer un potentiel d'action. Plusieurs types de points d'équilibres peuvent être représentés sur le plan de phase. Le *saddle node* et la bifurcation de hopf. La bifurcation de Hopf montre la variation de la stabilité d'un point d'équilibre.

2.4.5 « Limit cycle » et analyse de bifurcations

Les équations de Hodgkin-Huxley peuvent être analysées par l'observation des trajectoires des nullclines mais aussi par l'affichage graphique du plan de phase qui représente une boucle fermée. Cette représentation est appelée « *limit cycle* » et montre la trajectoire que prendra le système observé à un jeu de paramètres donné. Les points de bifurcations de Hopf peuvent être représentés par l'intersection de la *nullcline* avec la courbe de *limit cycle* afin de marquer le passage du système d'un état stable à un état instable ou inversement.

Les simplifications ont été faites de telle manière que les nouvelles équations du modèle possèdent une dynamique proche des originaux. L'analyse de bifurcations permet d'étudier la réponse du système à une perturbation, un stimulus, soit, dans notre cas un courant de stimulation appliqué au modèle [IZHI 2000] [RINZ 1989]. La figure 2.6 représente l'analyse de bifurcations des modèles optimisés, proposés dans les parties précédentes, et le modèle de Hodgkin-Huxley original via les paramètres de [POSP 2008]. Plusieurs observations peuvent être apportées. La dynamique des analyses de bifurcations de A et B est légèrement différente. Cependant, la plage de valeurs de courants de stimulation utilisée dans le cadre des expériences biologiques est $[0\,;50]$ $\mu A/cm^2$ sur laquelle l'analyse de bifurcations montre que la dynamique du système reste similaire.
Une différence peut être observée sur la première bifurcation de Hopf. En effet, les *nullclines* sont différentes, possiblement causée par le calcul des tangentes hyperboliques. Ces points représentent le moment où le système passe d'un état stable à un état instable. Il en résulte une fréquence de *spiking* plus élevée pour le neurone simplifié. Il est possible de compenser cette différence en créant une fonction non linéaire du courant de stimulation permettant de faire apparaître les mêmes fréquences de *spiking*. Les équations 2.23 et 2.24 représentent

les fonctions des courants à appliquer sur le neurone simplifié pour obtenir la même dynamique. Ils ont été déterminés par régression linéaire à partir de plusieurs points afin de faire correspondre les courbes F-I.

$$f_{FS}(I_s) = (7,58I_s) + 3,381 \tag{2.23}$$

$$f_{RS}(I_s) = (4,3I_s) + 0,66 \tag{2.24}$$

avec f_{FS} et f_{RS} les fonctions dépendant de Is, le courant de stimulation, permettant de compenser la différence apportée par les simplifications.

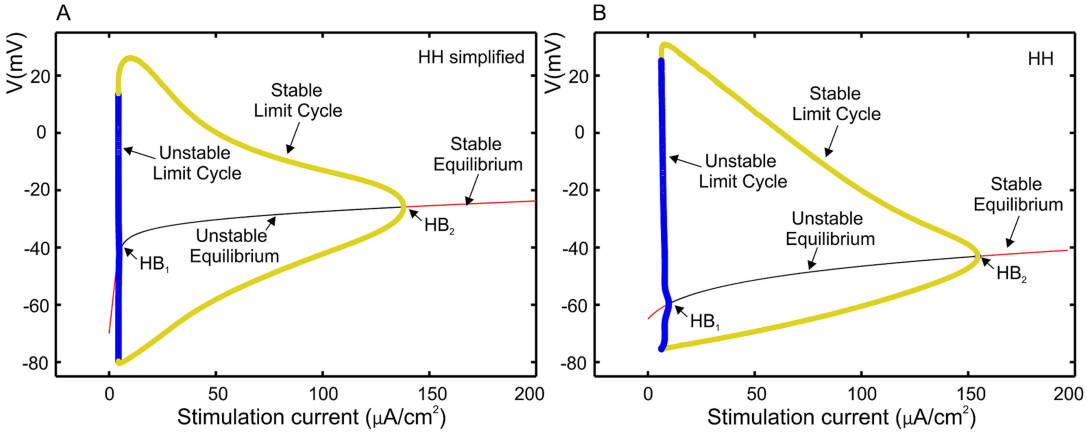

FIGURE 2.6 – Comparaison des modèles en utilisant l'analyse de bifurcations : A) Analyse de bifurcations du modèle optimisé; B) Analyse de bifurcations du modèle original de Hodgkin-Huxley.

2.5 Modèles à compartiments

Le chapitre 1 a présenté succinctement le fonctionnement d'un modèle à compartiments. Cette section a pour but de détailler les équations présentées par [HAY 2011] qui est le modèle retenu et d'expliquer la manière dont le modèle a été implémenté sur FPGA.

2.5.1 Èquations du modèle

Les méthodes citées précédemment peuvent être appliquées aux équations de la table 2.7.

Ainsi, les équations originales présentées par [HAY 2011] ont été modifiées afin d'être optimales à une implémentation numérique en utilisant les méthodes montrées précédemment. Les courants ioniques mettant en relation plusieurs types de canaux ioniques et sont représentés par l'équation 2.25.

$$I_x = g_x m^p h^q (V - E_x) \tag{2.25}$$

avec, I_x le courant ionique associé aux ions x, g la conductance, m et p respectivement les fonctions d'activation et d'inactivation, V_m la tension de membrane et E_x le potentiel d'équilibre. Les équations des courants ioniques ainsi que les ordres des puissances p et q des fonctions d'activation et d'inactivation sont représentés sur le tableau 2.7.

Courants	p	q	Equations
Fast-inactivating Na+ (INat)	3	1	$\alpha_m = \frac{0.182(V+38)}{1-e^{-\frac{V+38}{6}}}$ $\alpha_h = \frac{-0.015(V+66)}{1-e^{-\frac{V+66}{6}}}$ $\beta_m = -\frac{0.124(V+38)}{1-e^{\frac{V+38}{6}}}$ $\beta_h = -\frac{0.015(V+66)}{1-e^{\frac{-(V+66)}{6}}}$
Persistant Na+ (INap)	3	1	$\alpha_m = \frac{0.182(V+38)}{1-e^{-\frac{V+38}{6}}}$ $\alpha_h = \frac{-2.88.10^{-6}(V+37)}{1-e^{-\frac{V+117}{4.63}}}$ $\beta_m = -\frac{0.124(V+38)}{1-e^{\frac{V+38}{6}}}$ $\beta_h = \frac{6.94.10^{-6}(V+64.4)}{1-e^{\frac{-(V+64.4)}{2.63}}}$
Non-specific cation (Ih)	1	0	$\alpha_m = \frac{0.00643(V+154.9)}{e^{-\frac{V+154.9}{11.9}}-1}$ $\beta_m = 0.00193.\mathrm{e}^{\frac{V}{33.1}}$
Muscarinik K+ (Im)	1	0	$\alpha_m = 0.0033.e^{0.1.(V+35)}$ $\beta_m = 0.0033.e^{-0.1.(V+35)}$
Slow inactivating K+ (IKp)	1	0	$m_\infty = \frac{1}{1+e^{-\frac{(V+11)}{12}}}$ $h_\infty = \frac{1}{1+e^{\frac{(V+64)}{11}}}$ $\tau_m = \frac{125+175.03.e^{\mp 0.026.(V+1}}{T_{adj}}$ $\tau_h = \frac{360+(24V+2594)e^{-\left(\frac{V+85}{48}\right)^2}}{T_{adj}}$
Fast inactivating K+ (IKt)	4	1	$m_\infty = \frac{1}{1+e^{\frac{-(V+10)}{19}}}$ $h_\infty = \frac{1}{1+e^{\frac{V+76}{10}}}$ $\tau_m = \frac{0.34+0.92e^{-\left(\frac{V+81}{59}\right)^2}}{T_{adj}}$ $\tau_h = \frac{8+49e^{-\left(\frac{V+83}{23}\right)^2}}{T_{adj}}$
Fast non inactivating K+	1	0	$m_\infty = \frac{1}{1+e^{\frac{-(V+10)}{19}}}$ $\tau_m = \frac{0.34+0.92e^{-\left(\frac{V+81}{59}\right)^2}}{T_{adj}}$
High voltage activated Ca2+ (IHVA)	2	1	$\alpha_m = \frac{-0.055(V+27)}{e^{\frac{-(V+27)}{3.8}}-1}$ $\alpha_h = 0.000457e^{\frac{-(V+13)}{50}}$ $\beta_m = 0.94e^{\frac{V+75}{17}}$ $\beta_h = \frac{0.0065}{e^{\frac{-(V+15)}{28}}+1}$
Low voltage activated Ca2+ (ILVA)	2	1	$m_\infty = \frac{1}{1+e^{\frac{-(V+40)}{6}}}$ $h_\infty = \frac{1}{1+e^{\frac{V+90}{6.4}}}$ $\tau_m = 5+\frac{20}{T_{adj}\,(1+e^{\frac{V+35}{5}})}$ $\tau_h = 20+\frac{50}{T_{adj}\,(1+e^{\frac{V+50}{7}})}$

FIGURE 2.7 – équations présentées par [HAY 2011].

La simplification amène une nouvelle fois à unifier les équations de la table 2.7 afin d'effectuer les calculs avec des constantes, des décalages de bits et des additions/soustractions. Ces nouvelles équations sont représentées à partir des équations 2.8 à 2.9 et des tables 2.7 et 2.8 contenant les constantes permettant de représenter les fonctions de canaux ioniques désirés. La table 2.8 correspond aux coefficients permettant de calculer les termes x_∞, avec x la représentation de m ou h, et la table 2.9 les coefficients pour le calcul de τx.

2.5.2 Description des courants ioniques

Il a été expliqué dans le chapitre 1 la complexité du fonctionnement des neurones par le grand nombre de canaux ioniques dont la mécanique complexe joue un rôle dans la dynamique des cellules. Pour rappel, plus de 200 types de canaux ioniques ont été découverts à

		a	b	c	d
I_{Nat}	m	2^{-1}	$2^{-4}+2^{-6}$	3,149	0,4997
	h	-2^{-1}	$2^{-4}+2^{-6}$	5,156	5,212
I_{Nap}	m	2^{-1}	$2^{-4}+2^{-5}$	4,932	0,4997
	h	2^{-1}	$2^{-5}+2^{-6}$	2,285	0,4988
I_h	m	2^{-1}	$-(2^{-5}+2^{-6})$	5,19	0,4994
I_m	m	2^{-1}	$2^{-4}+2^{-5}$	3,282	0,4998
I_{Kp}	m	2^{-1}	$2^{-5}+2^{-6}$	0,5155	0,5
	h	2^{-1}	$2^{-5}+2^{-6}$	3	0,501
I_{Kt}	m	2^{-1}	$2^{-5}+2^{-6}$	0,5155	0,5
	h	2^{-1}	-$(2^{-5}+2^{-6})$	-3,561	0,4987
I_{HVA}	m	2^{-1}	$2^{-4}+2^{-5}$	3,106	0,4953
	h	2^{-1}	-$(2^{-6}+2^{-7})$	-1,4	0,5188
I_{LVA}	m	2^{-1}	$2^{-4}+2^{-5}$	3,751	0,4997
	h	2^{-1}	-$(2^{-4}+2^{-6})$	-7,031	0,5

TABLE 2.7 – Table de coefficients pour le calcul de x_{inf} en fonction des types de courant ioniques désirés.

		a	b	c	d	e
INat	m	$-1,501\times10^{-3}$	$9,455\times10^{-2}$	$6,255\times10^{-2}$	2,443	$2,87\times10-2$
	h	$5,08\times10^{-2}$	5,212	$6,6\times10^{-2}$	4,891	$2,217\times10^{-1}$
INap	m	$-9,012\times10^{-3}$	$5,671\times10^{-1}$	$6,255\times10^{-2}$	2,443	0,1722
	h	-	-	-	-	-
I_h	m	$5,563\times10^{-1}$	141	$3,676\times10^{-2}$	4,107	$3,635\times10^{-1}$
I_m	m	$3,562\times10^{-9}$	51,31	-0,1	-3,5	$-3,342\times10^{-7}$
I_{Kp}	m	-	-	-	-	-
	h	2,895	498,2	$-4,24\times10^{-2}$	-2,773	106,9
I_{Kt}	m	$5,056\times10^{-2}$	16,06	$4,607\times10^{-2}$	2,867	1,667
	h	$-3,08\times10^{-1}$	-9,285	$6,205\times10^{-2}$	4,799	3,151
I_{HVA}	m	$-1,102\times10^{-1}$	2,33	$9,341\times10^{-2}$	3,469	$3,727\times10^{-1}$
	h	$-3,08\times10^{-1}$	-9,285	$6,205\times10^{-2}$	4.799	3.151
I_{LVA}	m	-	-	-	-	-
	h	$-3,538\times10^{-1}$	-7.733	$5,177\times10^{-2}$	3,322	7,061

TABLE 2.8 – Table de coefficients pour les calculs de τ_x.

ce jour. De ce fait, les deux types de canaux ioniques (sodium et potassium) présents dans le modèle de Hodgkin-Huxley original ne sont pas suffisants. En effet, il a aussi été expliqué dans le chapitre précédent que les maladies neurologiques peuvent affecter les canaux ioniques. De plus, ces derniers jouent un rôle dans la dynamique complexe de la génération des influx nerveux. La figure 2.7 représente les équations permettant de calculer plusieurs types de courants ioniques. Chacun d'entre eux possède un rôle dans les différents compartiments du modèle.

2.5.3 Effet des compartiments

Les avantages de l'utilisation d'un modèle à compartiments sont multiples. [DAVI 2005] a montré la précision que les compartiments apportent sur le potentiel d'action. Outre cette précision qui peut être sujette à discussion sur son utilité, d'autres phénomènes présents dans les neurones biologiques, mais non représentés sur le modèle de HH original, peuvent être apportés. Parmi ces phénomènes, on peut citer les *spikes* dendritiques. Il a été expliqué dans le chapitre 1 que les dendrites n'étaient pas toujours passives, ainsi que l'importance de leur rôle dans la mémoire et l'apprentissage, notamment sur le phénomène de potentiation à long terme. Dans certains cas et certaines conditions, ils peuvent, comme le segment initial de l'axone, générer un potentiel d'action. Il a été montré que les dendrites possèdent des canaux ioniques, et, sont capables de générer des signaux électriques dans plusieurs parties du cerveau [KENDE 1960]. Un type particulier de *spike* apparaît dans les dendrites dont les récepteurs sont sensibles aux neurotransmetteurs NMDA et sont appelés dans la littérature « NMDA *spike* ». Ce mécanisme n'est pas encore bien décrit mais, selon [SCHIL 2000], il jouerait un rôle dans la potentiation à long terme des synapses. Un autre phénomène survient lorsqu'un potentiel d'action se propage vers un soma. Ce dernier génère alors d'autres *spikes* qui seront propagés sur les dendrites. Ce phénomène est appelé *backpropagation* et jouerait un rôle sur les potentiations à court terme. En outre, il a été montré que cette propagation des potentiels d'action à travers les dendrites entraînait une augmentation du calcium dans les régions distales-apicales des dendrites et une augmentation du sodium dans les régions périsomatiques. Ces variations d'ions entraînent deux autres phénomènes respectivement appelés CA2+ *spike* et Na+ *spike*.

Le modèle proposé par [HAY 2011] reproduit les activités électriques d'une cellule pyramidale de la couche 5b du néocortex. Il s'agit de la couche la plus profonde dont les dendrites des neurones s'étendent sur toute la colonne en recevant des informations de toutes les couches. Ces neurones ont aussi la particularité d'accéder à plusieurs parties du cerveau. Le modèle en question reproduit les deux types de *spikes* cités précédemment.

2.5.4 Implémentation

Le modèle de neurone choisi ici représente les dendrites apicales, les dendrites basales, l'axone et le soma. Selon le type de compartiment, certains courants ioniques ne seront pas utilisés. Par exemple, la dendrite basale n'utilise que les courants h et fuite. Il est possible d'utiliser le même cœur de calcul pour représenter tous les compartiments. Les courants ioniques non utilisés auront des valeurs de conductances à 0. Les paramètres des conductances et des potentiels d'équilibre sont représentés sur le tableau 2.9.

Chaque compartiment est représenté par le formalisme de Hodgkin-Huxley. Il est possible de partir du principe que les compartiments sont des neurones connectés. Ainsi, un

		Soma	D. Apical	D. Basal	Axone
Conductances G (en S)	Nat	2,41114	$2,13.10^{-2}$	0	0
	Nap	$2,06183.10^{-3}$	0	0	0
	h	$2,79218.10^{-4}$	4.10^{-4}	$2,935.10^{-4}$	0
	m	0	$6,75.10^{-5}$	0	0
	Kp	$1,75889.10^{-4}$	0	0	0
	Kt	$2,38758.10^{-2}$	0	0	0
	Kv	0.700652	0	0	0
	CaHVA	$8,38171.10^{-4}$	$5,55.10^{-4}$	0	0
	CaLVA	$3,11435.10^{-3}$	$1,87.10^{-2}$	0	0
	SK	$4,78551.10^{-2}$	$2,61.10^{-4}$	0	0
	Leak	$4,371.10^{-6}$	$5,89.10^{-5}$	$3,7935.10^{-6}$	$4,56.10^{-5}$
Potentiels D'équilibre E (en mV)	K	-85	-85	-85	-85
	Na	50	50	50	50
	Leak	-90	-90	-90	-90
	h	-40.6682	-40.6682	-45	-45
	Ca	-132.46	-135	-126	-135

TABLE 2.9 – Paramètres de conductances et de potentiels d'équilibre pour chaque compartiment issus du modèle simplifié proposé par [MAKI 2018].

neurone à compartiments est une matrice de neurones dont les paramètres diffèrent. Le résultat de l'implémentation du modèle optimisé est visible sur la Figure 10. La figure 10.a) représente la morphologie du neurone simulé et fait aussi office de légende. En effet, les couleurs des sondes sont celles utilisées dans les graphiques b, c et d. La figure 10.b) montre le comportement du neurone lorsqu'un courant constant est injecté sur le soma et affiche un comportement proche de la biologie. [HAY 2011] décrit les neurones de la couche L5b par une activité débutant par une bouffée de *spikes* et continuant ses activités avec une fréquence constante. La figure 10.c) représente la propagation d'un potentiel d'action somatique dans les dendrites. La figure 10.d) montre, lors d'une stimulation plus intense, un *spike* dendritique.

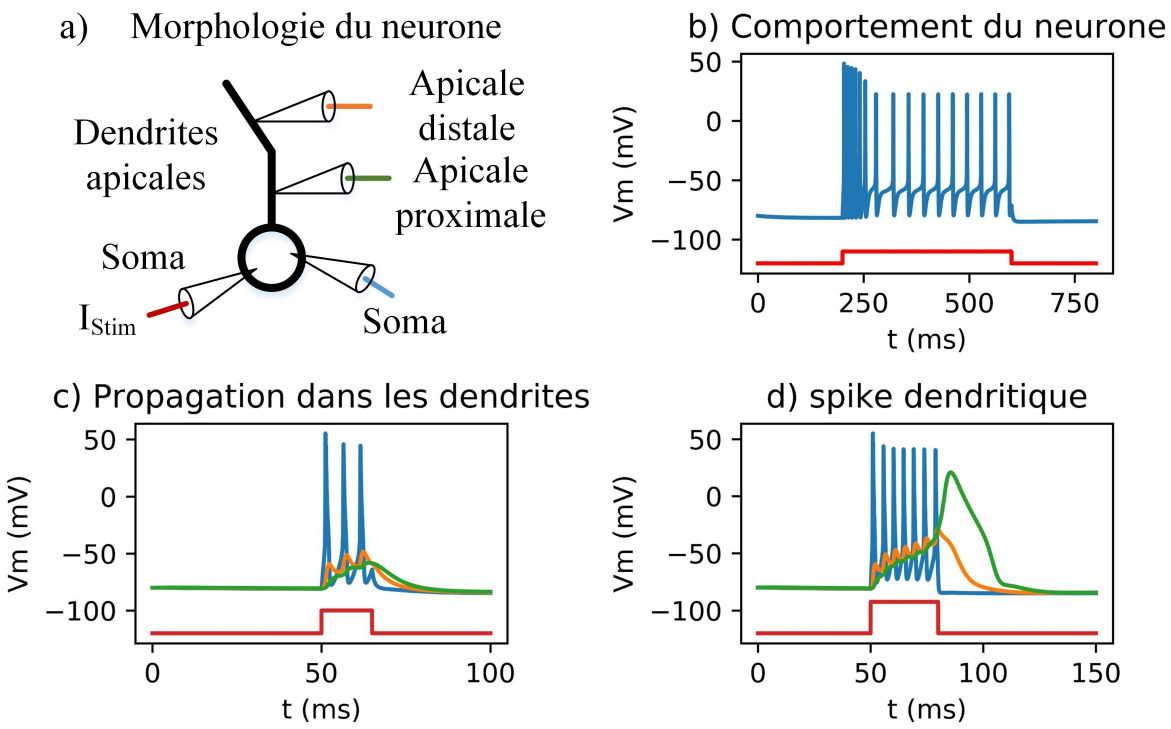

FIGURE 2.8 – Résultat de l'implémentation post-implémentation et comparaison avec le logiciel Neuron : a) soma; b) Dendrite apicale montrant une atténuation et un déphasage temporel par rapport à la tension de membrane du soma. La stimulation s'effectue sur le neurone et la tension se propage à travers les dendrites; c) Il est possible d'observer sur une dendrite apicale proximale (orange) la propagation de la tension de membrane lors de l'application d'une courte stimulation I_{stim} plus forte que celle appliquée sur b). De plus, des *spikes* dendritiques apparaissent sur la dendrite apicale distale; d) En augmentant l'intensité et la durée de I_{stim}, les tensions présents sur les dendrites augmentent. Lorsque la tension de membrane de la dendrite apicale proximale atteint un certain seuil, un *spike* apparaît (vert).

La relation entre un compartiment et un autre est liée par leurs structures qui, dans le cas des modèles à compartiments, sont représentées par des résistances axiales. Le courant que recevra un compartiment d'un autre compartiment peut donc être résumé par les équations 2.26 et 2.27.

$$I_a = \frac{\Delta_V}{r_m L} \tag{2.26}$$

$$r_m = R_a / 2\pi D_a \tag{2.27}$$

avec ΔV la différence de tension entre le compartiment considéré et celui qui y est connecté, r_m la résistance de membrane et L la longueur du compartiment, R_a la résistance spécifique de la membrane et D_a le diamètre du compartiment.

Les équations 2.26 et 2.27 montrent la relation qui lie un compartiment à un autre. Lorsque plusieurs compartiments sont connectés à un seul compartiment, il faudra calculer la somme des courants axiaux. Si l'on considère une représentation des compartiments sous forme d'une matrice de connectivité, il est alors possible d'effectuer un produit matrice

vecteur pour obtenir un vecteur contenant les courants axiaux à injecter dans chaque compartiment. Il y a plusieurs avantages à utiliser les calculs matriciels. Les calculs seront facilités ce qui impactera la vitesse de calcul et induira une réduction des ressources à utiliser sur le FPGA. Aussi, l'utilisation d'une matrice peut permettre la reproduction de morphologie en trois dimensions. Cependant, l'inconvénient est que le nombre de données à utiliser sera égal au nombre de compartiments au carré. La figure 2.9 montre différentes configurations de matrices possibles pour générer différentes morphologies. Plusieurs types de neurones peuvent ainsi être simulés.

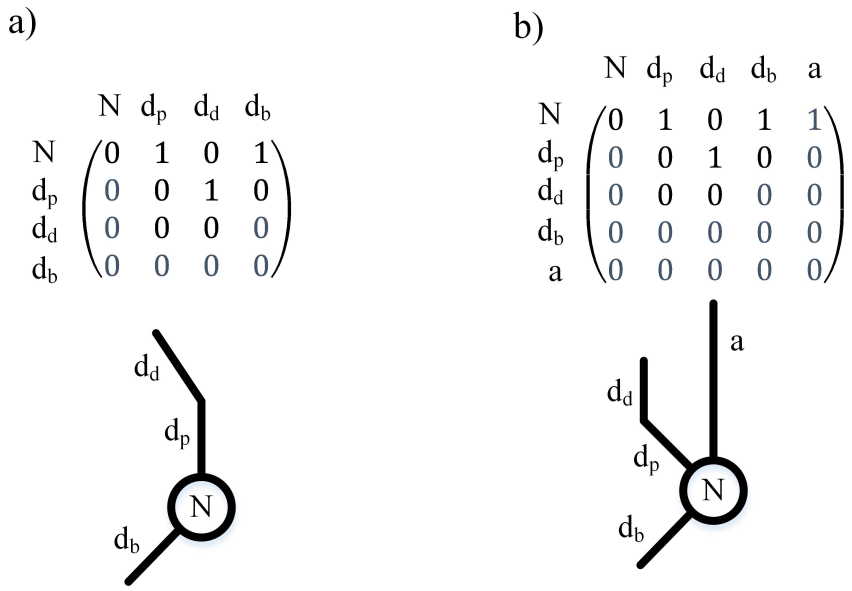

FIGURE 2.9 – Représentation des compartiments sous forme de matrices. N représente le neurone, d_p la dendrite apicale proximale, d_d la dendrite apicale distale, d_b la dendrite basale et a l'axone : a) représente la morphologie décrite par le modèle à quatre compartiments de [HAY 2011] optimisé par [MAKI 2018] ; b) Le neurone possède un axone supplémentaire.

2.6 Résumé

Ce deuxième chapitre a permis tout d'abord de présenter le contexte technologique en exposant les différentes solutions présentes dans la littérature afin de simuler des réseaux de neurones ainsi que différentes méthodes de simplification des équations. Les méthodes mathématiques permettant d'optimiser les équations de HH tout en conservant les paramètres biophysiques sont les méthodes de régression et les méthodes d'interpolation. Plusieurs études montrent une application concrète de ces méthodes appliquées aux modèles de neurones. Cependant, ces méthodes ont des applications restreintes sur le type de neurones utilisés et le nombre de neurones pouvant être simulés. Ainsi, une méthode optimisée pour l'implémentation numérique a été créée dans le cadre des travaux de thèse. Cette méthode utilise la régression et l'interpolation afin de générer des équations génériques dont les paramètres dépendent des types de neurones. L'utilisation d'une même équation permet de limiter les ressources du FPGA. Les résultats de l'application de cette méthode montrent de bons résultats sur l'implémentation de neurones et des synapses. L'implémentation des synapses faisant l'objet d'un outil supplémentaire : le produit d'une matrice par un vecteur. Pour aller plus loin dans l'implémentation, un courant bruité a été ajouté afin de simuler les

activités spontanées, dont l'importance a été montrée dans le chapitre 1. Le nombre d'activités spontanées par seconde peut être modifié en changeant les paramètres de l'équation utilisée (Ornstein-Uhlenbeck). Cette possibilité a été démontrée en comparant avec plusieurs études. La variabilité de la dynamique du modèle de neurone induite par l'optimisation peut être étudiée par plusieurs outils. Le Pearson Product Moment Correlation (PPMC) permet de quantifier la différence entre deux signaux lorsque la Cross-Correlation permet d'ajouter une dimension temporelle, c'est-à-dire, une différence de phase. La dynamique des neurones peut aussi être étudiée en observant les courbes F-I et le plan de phase généré par une analyse de bifurcations. Enfin, une implémentation d'un modèle à compartiments a permis de valider les outils mis en place mais aussi l'ajout de phénomènes bioréalistes aux neurones artificiels.

Chapitre 3

De la bibliothèque VHDL aux systèmes pour l'ingénierie neuromorphique

3.1 Introduction

Le chapitre 2 a décrit plusieurs méthodes permettant d'adapter les équations de Hodgkin-Huxley en vue d'une implémentation numérique. Chaque méthode a été définie suivant les contraintes techniques du FPGA en considérant les facteurs de temps, de vitesse et de ressources (LUT, Bascules D, DSP, mémoire). La particularité du FPGA est qu'il offre la possibilité de contrôler chaque composant qu'il inclut afin de réaliser l'architecture désirée. Ce chapitre a pour but d'expliquer les choix techniques qui ont été faits afin de répondre aux contraintes de temps et de ressources. Ces choix incluent des méthodes et algorithmes de calcul mais aussi des techniques d'optimisation des ressources. Ainsi, une bibliothèque en VHDL a été réalisée pour encoder les informations de manière optimale mais aussi afin de réaliser des fonctions et des opérations communément utilisées dans l'ingénierie neuromorphique. Enfin, ce chapitre montrera le système de transmission d'informations mis en place entre le FPGA et l'ordinateur afin de modifier la configuration des neurones en temps réel. Ce chapitre conclura sur les architectures générales des réseaux de neurones et de la mise en pratique de toutes les techniques et méthodes présentées.

3.2 Codage et précision

Ce chapitre débute avec le codage des nombres dans le FPGA, car il s'agit d'une partie importante et déterminante sur les choix des circuits qui seront présentés par la suite. Le FPGA, comme tout système numérique, représente les valeurs sous la forme d'un codage binaire. Une valeur entière sera donc représentée par l'équation 3.1.

$$\sum_i b_i 2^i \tag{3.1}$$

Le problème de cette représentation est qu'elle ne peut représenter que les valeurs entières positives dites « valeurs non signées ». Les valeurs négatives, soit dites signées, peuvent être représentées par l'ajout d'un bit qui déterminera le signe. L'opération consistant à convertir une valeur positive en valeur négative est appelée le complément à 2. Si A est une valeur entière positive alors $-A = \bar{A} + 1$ avec $\bar{A}$ l'inversion de tous les bits. En partant de cette base, il existe deux façons de représenter les valeurs réelles. La première est la représentation par virgule flottante qui est majoritairement utilisée dans les ordinateurs, supercalculateurs et systèmes embarqués à base de processeurs. Les flottants sont représentés par un signe, une mantisse ou significande, une base et un exposant. L'équation 3.2 montre comment est représentée une valeur en base 10 sous forme de virgule flottante.

$$(x)_{10} = smb^e \tag{3.2}$$

Avec x une valeur réelle de base 10, s le signe, m la mantisse, b la base et e l'exposant. Les vecteurs de bits sont décalés, d'où l'appellation virgule flottante, vers la gauche ou la droite selon la valeur de l'exposant.
Ce type de codage est complexe et coûteux en ressources et temps de calcul à implémenter dans un système tel que le FPGA. Une autre représentation à base de virgule fixe existe et permet une implémentation plus simple. Cependant, la précision peut être impactée selon le choix du nombre de bits. Si l'on prend l'équation 3.3 représentant l'équivalence entre un nombre de base 10 et son calcul à partir de sa représentation binaire sous forme de virgule fixe.

$$(x)_{10} = \sum_i bi_i.2^i + \sum_j bd_j.2^{-j} \tag{3.3}$$

avec x, une valeur réelle de base 10, *bi* et *bd* sont respectivement la représentation binaire de la partie entière et de la partie décimale de x sous forme de vecteurs de bits, i et j sont les indices des vecteurs.

Le codage sous forme de virgule fixe a ainsi été choisi pour plusieurs raisons. La première est que les travaux précédents de l'équipe ont fait leurs preuves [AMBR 2013][AMBR 2015] et ont montré que la précision générée par une partie décimale de 18 bits était suffisante pour générer un potentiel d'action à partir des équations de Izhikevich. De plus, l'implémentation ainsi que les opérations binaires de ce type de codage sont bien plus simples. Ainsi, une bibliothèque VHDL de virgule fixe a été créée. Cependant, tous les opérateurs ne sont pas implémentés dans les bibliothèques et pour celles qui le sont, elles ne sont pas forcément optimales ou adaptées au système désiré. La suite du chapitre détaillera donc le développement d'opérateurs nécessaires à la conception de systèmes neuromorphiques en VHDL et partira sur la base d'opérations sur des valeurs encodées en binaires signé sous forme de virgule fixe.

3.3 Temps de propagation et pipeline

Il existe plusieurs types d'algorithmes, à savoir, parmi ceux qui nous intéressent, les algorithmes itératifs et multiplicatifs cités par [OBER 1995], pour réaliser les fonctions désirées. Plusieurs contraintes doivent être prises en compte afin d'adapter un algorithme ou une fonction à une utilisation sur FPGA. Il faut distinguer les algorithmes itératifs des autres d'une part et, d'autre part, étudier la faisabilité de l'implémentation en fonction des ressources disponibles/souhaité,s et de la vitesse (fréquence de fonctionnement). Le FPGA est principalement utilisé pour du calcul synchrone, c'est-à-dire que chaque opération est synchronisée à un signal d'horloge. Ainsi, la vitesse d'un système est déterminée par le nombre de cycles d'horloge qu'il mettra pour fournir un premier résultat. Pour réduire ce temps de calcul, il est possible de réaliser un certain nombre d'opérations entre deux cycles d'horloge au risque d'induire un temps de propagation trop important. Ces notions de cycle d'horloge, d'opération et de temps de propagation seront développées dans l'annexe 2.

L'optimisation des systèmes présentés dans ces travaux de thèse passe par un équilibre entre les temps de propagation, les circuits pipeline et l'utilisation de circuits dédiés. L'importance de ces critères affecte tout d'abord la fréquence d'horloge, ce qui impacte explicitement la vitesse du système et donc le nombre de calculs qui pourront être effectués par unité de temps. Ensuite, il est important de prendre en compte les temps de propagation afin d'avoir un système stable et fonctionnel. Enfin, l'utilisation des circuits dédiés permet un fonctionnement optimal du système avec une utilisation réduite des ressources du FPGA.

3.4 Multiplication et Division

3.4.1 Multiplication

Les premiers algorithmes qui sont encore utilisés aujourd'hui se basent sur les méthodes « papier et stylo » (*pencil and paper method*) de multiplication. Si le nombre à multiplier est

le multiplicande et que le nombre qui multipliera est le multiplieur, alors il est possible de calculer le résultat de la multiplication en multipliant chaque chiffre composant le multiplieur multiplié par le multiplicande comme dans la figure 3.1.a) où X1 et X2 représentent des éléments d'un vecteur X représentant le multiplicande et Y le multiplieur. Il peut s'agir d'un vecteur de bits comme d'un vecteur contenant les valeurs du tableau de numération des nombres à calculer. Le circuit numérique représentant le calcul de la figure 3.1.a) est représenté par le circuit de la figure 3.1.b).

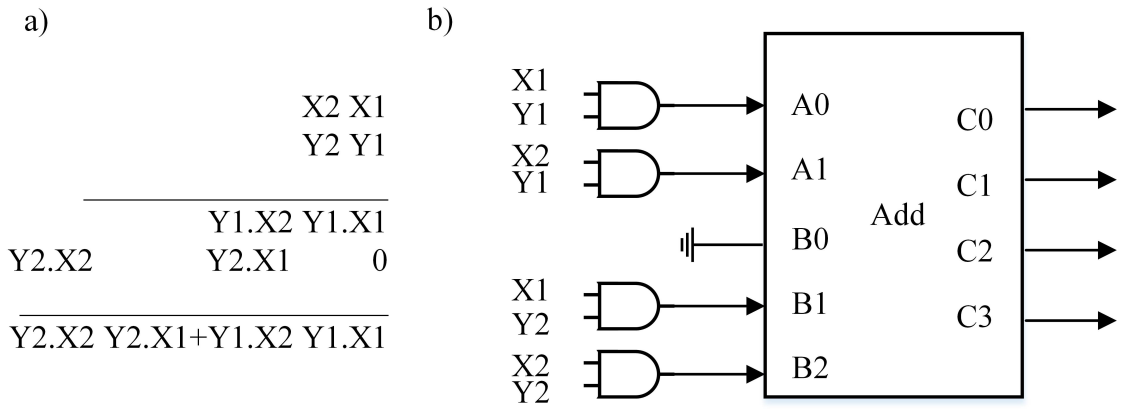

FIGURE 3.1 – Multiplication par algorithme itératif : a) algorithme de multiplication classique ; b) transposition de la multiplication classique pour application au codage binaire.

Cette solution permet de produire, à chaque étape, un bit, ce qui rend les performances de l'algorithme, dépendant de la taille des vecteurs à multiplier, peu efficaces. Il devient fiable lorsque le multiplieur est par exemple sur 2 bits (comme dans la partie consacrée au produit matrice vecteur) cependant, dans le cas du calcul des équations de Hodgkin-Huxley, le multiplicande et le multiplieur sont de tailles égales. Plusieurs autres algorithmes permettent d'augmenter les performances en réduisant la taille du chemin de transmission par l'ajout de calculs parallèles. Les architectures comme l'arbre de Wallace (*Wallace tree multiplier*) ou l'algorithme de booth, présentées par [MULL 2016], permettent d'augmenter les performances des multiplieurs mais, restent tout de même coûteuses en ressources. Le *Coordinate Rotation Digital Computer* (CORDIC), qui sera expliqué dans la prochaine partie, est un algorithme qui permet aussi de réaliser des multiplications mais est aussi coûteux que les autres et est limité dans son domaine de définition [VOLD 1959].
Une autre manière de procéder à une multiplication est d'utiliser un composant dédié présent dans certains FPGA appelé le *Digital Signal Processing* (DSP). Ce composant permet en un cycle d'horloge, avec une horloge inférieure à 100 MHz, de réaliser une multiplication entre une valeur de 18 bits et une autre de 24 bits. Le circuit offre aussi la possibilité d'augmenter la fréquence de fonctionnement du design en ajoutant des *pipelines* en entrée et en sortie. La figure 3.2 représente le schéma du fonctionnement d'un DSP. Celui-ci représente un DSP48E2 présent dans les modèles Ultrascale de chez Xilinx avec une possibilité d'avoir un à deux étages de *pipeline* en entrée, un pré-additionneur/soustracteur, un à deux étages de *pipeline* en sortie avec un post-additionneur/soustracteur. Il est tout aussi possible de n'utiliser que les additionneurs/soustracteurs à haute vitesse. La documentation de Xilinx [XILI 2019] indique que ces composants peuvent atteindre un peu plus de 500 MHz.

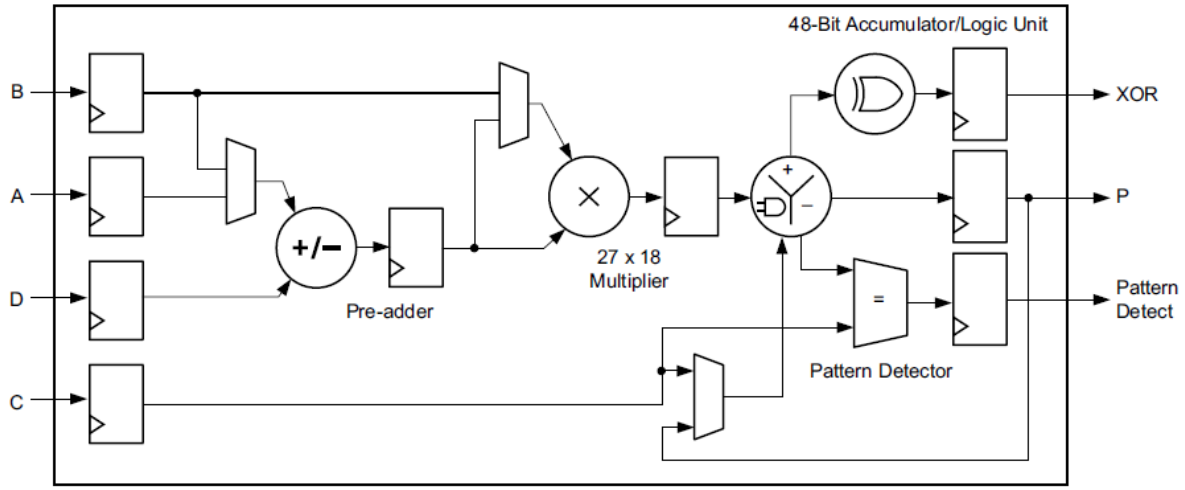

FIGURE 3.2 – Schéma du composant DSP48E2 issu de la documentation de Xilinx UG579 [XILI 2019] montrant les étages de *pipeline* avec un pré-additionneur/soustracteur, l'étage de multiplication, la possibilité d'avoir un post-additionneur/soustracteur, l'optimisation du mode de cascade des DSP, un détecteur de pattern pour, par exemple, les *overflows* ou *underflows* et un XOR sur vecteurs de 48 bits.

Les modèles DSP48E1 présents dans le Kintex 7 de la Genesys 2 sont tout aussi efficaces mais ne possèdent pas l'opérateur « ou exclusif » ou XOR en sortie du multiplieur et sont moins optimisés pour la mise en cascade des DSP ou la détection de *patterns*. Les détecteurs de *patterns* sont utiles pour mettre en place un détecteur d'*overflow* ou d'*underflow*. Lorsqu'un nombre atteint son maximum et qu'une opération sans détecteur d'*overflow* est effectuée sur celui-ci, la valeur devient incertaine. D'où l'importance d'avoir un détecteur de *patterns* qui prendra beaucoup moins de ressources qu'une méthode à base de comparateurs suivis d'un multiplexeur pour un détecteur classique reproduit via un circuit logique. Un autre avantage des DSP, c'est qu'ils peuvent être connectés en « cascade ». Il est possible de connecter un DSP à un autre pour effectuer des opérations entre eux. Ils sont pratiques pour additionner les résultats des multiplications sans ajouter de LUT ou d'utiliser des unités arithmétiques et logiques (ALU) supplémentaires. En revanche, le désavantage de l'utilisation du multiplieur est que tous les FPGA ne le possèdent pas, ce qui implique qu'un système conçu pour utiliser des DSP48 ne sera pas un système compatible avec tous les FPGA. Ce composant est un des plus pratiques dans un FPGA et sera utilisé dans toutes les fonctions qui seront expliquées par la suite pour tous les modes présentés.

3.4.2 Division

La division est une opération fréquente qui peut être retrouvée dans plusieurs équations. Les équations de Hodgkin-Huxley n'en font pas abstraction. Plusieurs cas spéciaux peuvent permettre de supprimer la division : lorsqu'un nombre est divisé par une constante, il est simplement possible de multiplier par l'inverse de la constante. Ou, si la division s'effectue par une puissance de 2, il est possible d'utiliser un simple décalage de bits. Pourtant, il existe beaucoup de cas où cette opération reste inévitable et où des algorithmes seront nécessaires comme dans le cas des calculs du terme τ_x des équations de HH qui fait partie des méthodes proposées dans le chapitre 2.

Comme pour la multiplication, il existe plusieurs algorithmes de division dont [OBER 1995] fait la description. Ils peuvent être séparés en deux catégories : les méthodes itératives et les méthodes multiplicatives. [OBER 1995] décrit la différence entre les deux types. La première utilise des additions/soustractions et converge vers le résultat de manière linéaire. Elle concerne les algorithmes restaurants, non-restaurants et SRT (sigle représentant les noms des créateurs : Sweeney, Robertson, et Tocher).

La deuxième méthode, quant à elle, utilise des multiplications afin d'avoir une convergence quadratique vers le résultat. Les algorithmes de Newton-Raphson et de Goldsmith sont les algorithmes multiplicatifs le plus souvent utilisés. Le problème des algorithmes à base de multiplications a été évoqué précédemment. Non seulement il faudrait plusieurs cycles d'horloge mais ce type de méthode prendrait en plus de précieux DSP, qui sont, pour rappel, limités dans le kintex 7 de la Genesys 2, et plusieurs LUT.
Au vu de la complexité d'implémentation des algorithmes, l'algorithme non-restaurant a été choisi pour sa simplicité et le fait que son implémentation prend peu de ressources. Il exploite la même méthode de division euclidienne classique mais en binaire. Á chaque itération, le dividende est divisé ou multiplié par 2 (décalage de bits) et additionné ou soustrait au diviseur selon le signe du reste qui prend initialement la valeur du dividende [ROBE 1958].
Le fait de n'utiliser que des additionneurs et des décalages de bits permet d'effectuer à haute vitesse. En effet, il est possible d'opérer une division en un cycle d'horloge à 50 MHz et en deux cycles à 100 MHz lorsque la taille du nombre ne dépasse pas les 21 bits, taille limite retrouvée grâce à des tests empiriques effectués jusqu'à l'apparition de temps de propagation anormalement long. Ce module de division sera utilisé pour créer d'autres fonctions et pour calculer les équations de HH.

3.5 Fonctions hyperboliques et exponentielles

Parmi les équations modifiées de Hodgkin-Huxley, deux opérateurs sont centraux. Le cosinus hyperbolique et la tangente hyperbolique. Par ailleurs, ces deux opérateurs sont souvent utilisés dans la littérature pour les équations de neurones [BONA 2014]. De plus, la fonction sigmoïde, qui peut être représentée par une tangente hyperbolique, est aussi souvent utilisée dans les calculs de réseaux neurones biomimétiques et bioinspirés (reconnaissance d'image, classification, etc.) [AHME 2017]. Ce dernier fait l'état de l'art des méthodes proposées dans la littérature afin d'implémenter ces opérateurs sur Hardware ou ne serait-ce que pour gagner du temps de calcul (FLOPS) sur des microprocesseurs peu puissants. La méthode la plus connue serait d'implémenter le développement en série de Taylor (équation 3.4) avec un ordre assez élevé pour que la propagation de l'erreur n'affecte pas les calculs. Le problème de ce calcul est qu'il nécessite, sur plusieurs itérations, des calculs de puissance et un calcul de division. Ainsi, on préférera s'orienter sur d'autres algorithmes moins coûteux et plus rapides.

$$tanh(x) = \sum_{i}^{N} \frac{2^{2i}(2^{2i}-1)B_{2i}x^{2-i}}{(2i)!} \tag{3.4}$$

avec B_{2i}, les nombres de Bernoulli.

3.5.1 CORDIC

Le *COordinate Rotation Digital Computer* (CORDIC) a été créé par [VOLD 1959] pour répondre à des problématiques similaires à celles présentes dans ces travaux, à savoir, d'optimisation numérique, de limitation de ressources et de vitesse de calcul. Il s'agit d'un algorithme itératif qui a donc besoin de plusieurs itérations pour donner son résultat et qui est seulement composé d'opérations d'additions, de soustractions et de décalages de bits. Le CORDIC permet de calculer les fonctions trigonométriques et hyperboliques.

Á partir du résultat d'un sinus et d'un cosinus hyperboliques, il est possible d'obtenir une tangente hyperbolique en passant par l'équation 3.5. L'algorithme de division a été optimisé pour fonctionner à 100 MHz et pour donner un résultat en deux cycles d'horloge. Cependant, plus la taille du vecteur sera grande et plus l'algorithme prendra de ressources. Il est aussi possible, pour réduire cet inconvénient, de séparer la partie décimale de la partie entière. En utilisant l'égalité de l'équation 3.6, il est possible de séparer la partie décimale de la partie entière en considérant cette dernière comme une constante. Il sera donc possible de stocker les valeurs de *tanh* de chaque valeur possible de la partie entière et de la sélectionner afin d'éviter d'effectuer des calculs supplémentaires.

$$tanh(x) = \frac{\sinh(x)}{\cosh(x)} \tag{3.5}$$

$$\tanh(a+b) = \frac{\tanh(a)+\tanh(b)}{(1+\tanh(a)\tanh(b))} \tag{3.6}$$

Le système qui découle de l'équation 3.6 est représenté sur la figure 3.3 et montre deux chemins parallèles représentant d'un côté le calcul via le module CORDIC de la tangente hyperbolique qui calcule la partie décimale de θ et d'un autre côté, la sélection de la case mémoire correspondant à la valeur de la tangente hyperbolique de la partie entière stockée en mémoire. Les deux chemins finissent par se rejoindre pour donner le résultat final. Les registres « reg » correspondent à des bascules D permettant de synchroniser les données, de rendre le système *pipeline*. Les multiplication s'effectuent via un DSP et les divisions via la méthode non-restaurante.

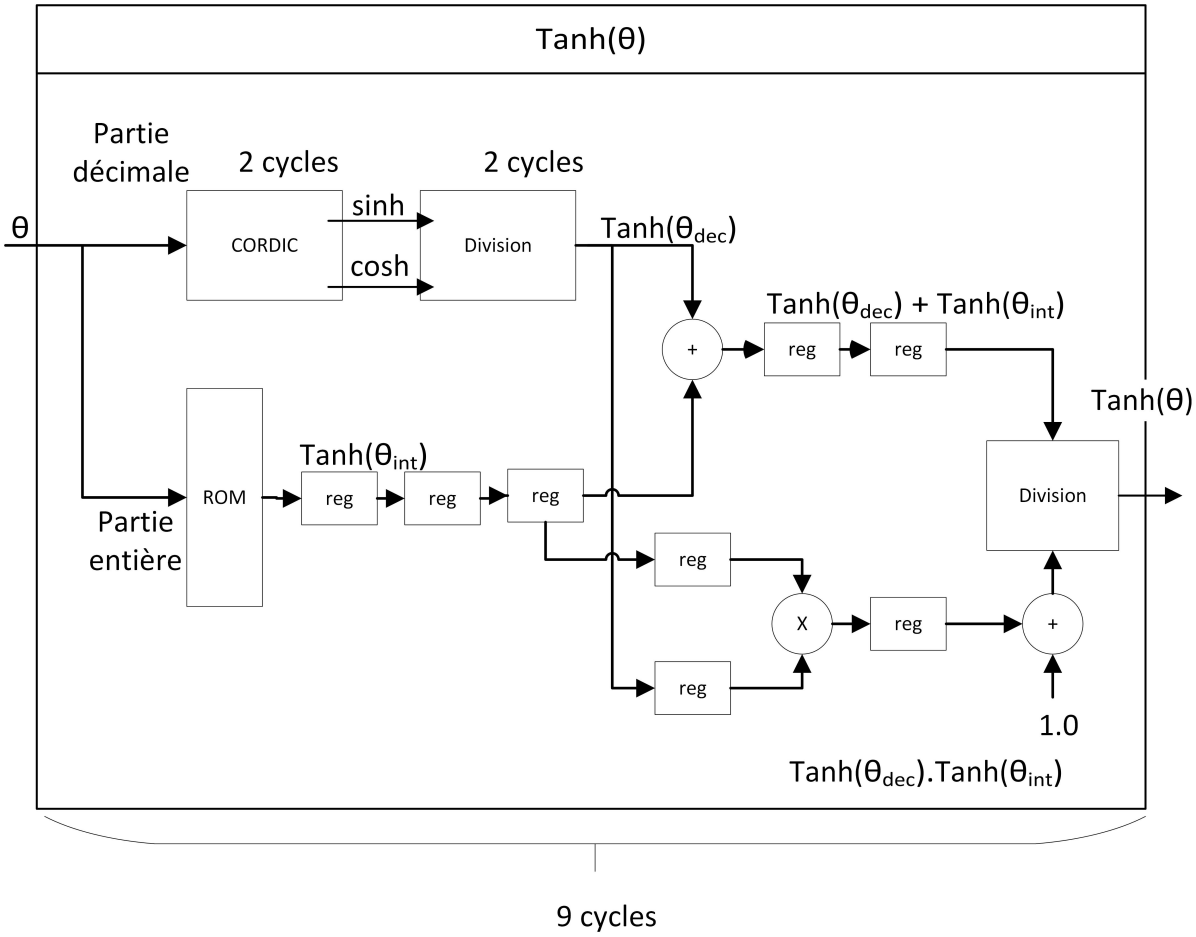

FIGURE 3.3 – architecture de la tangente hyperbolique en utilisant le CORDIC et les modules de division non-restaurantes.

D'autres déclinaisons des fonctions hyperboliques permettent au CORDIC de réaliser les fonctions cosinus hyperbolique (équation 3.7) et exponentielles (équation 3.8).

$$\cosh(x_{int} + x_{dec}) = \cosh(x_{int})\cosh(x_{dec}) + \cosh(x_{int})\cosh(x_{dec}) \tag{3.7}$$

$$e^{x} = \cosh(x) + \sinh(x) \tag{3.8}$$

avec x la valeur à calculer, x_{int} la partie entière de x et x_{dec} la partie décimale de x.

3.5.2 Interpolation de Lagrange

Le CORDIC est un algorithme itératif flexible permettant d'effectuer plusieurs opérations. Cependant, la question du calcul de la sigmoïde dans les réseaux de neurones a été largement étudiée et a amené à considérer l'utilisation des techniques d'interpolation [AHME 2017]. Une des méthodes d'interpolation la plus simple est l'interpolation de Lagrange qui est représentée par l'équation 3.9.

$$L(x) = y_0 + \frac{y_1 - y_0}{x_1 - x_0}(x - x_0) \tag{3.9}$$

avec L(x) la fonction d'interpolation de Lagrange, [x0, y0] les coordonnées du point directement situé avant le résultat, [x1, y1] les coordonnées du point directement situé après le

résultat. Dans le cas de la tangente hyperbolique, si l'on considère une interpolation sur N points, le résultat de tanh(x) sera situé entre les points choisis pour l'interpolation. La figure 3.4 représente le résultat d'une interpolation sur 10 points espacés de 0,5 de la tangente hyperbolique. Les 10 points choisis sont strictement positifs, car il suffit de multiplier L(x) par le signe de x car *tanh* est une fonction paire.

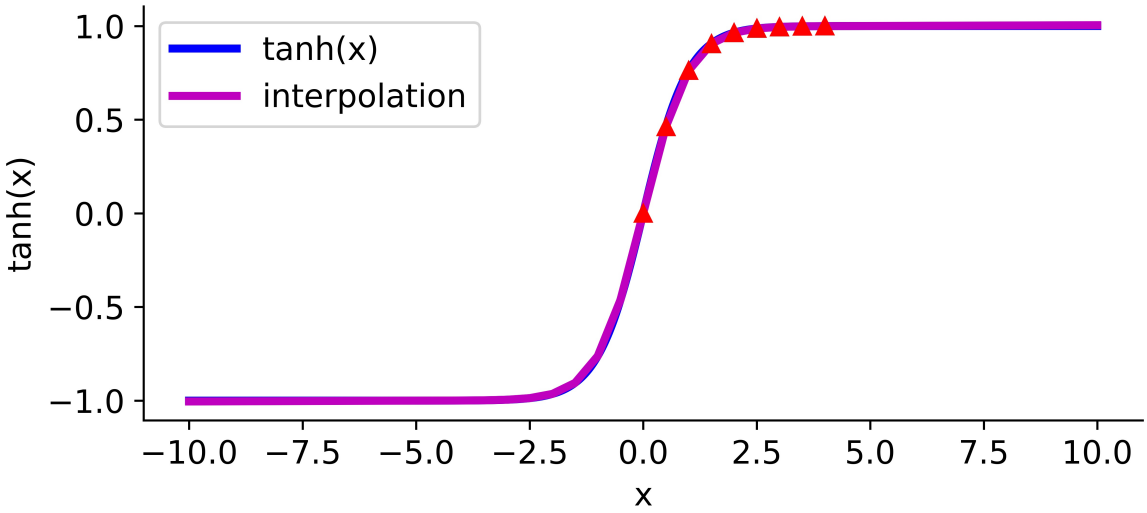

FIGURE 3.4 – Représentation de l'interpolation de Lagrange. La courbe rouge est le résultat de l'interpolation, sur 10 points qui sont représentés par des triangles rouge sur la courbe

L'utilisation d'un nombre limité de points (10) permet de limiter l'utilisation de la mémoire. De plus, lorsque les points sont choisis, $\frac{y_1-y_0}{x_1-x_0}$ est une constante. Ainsi, l'implémentation numérique consistera à stocker les constantes en mémoire et à enregistrer le signe de x à chaque calcul, puis de multiplier la constante par le terme $x-x0$. Ce calcul peut être effectué par un DSP en utilisant l'entrée inverseurs, qui permet de changer le signe du résultat, la fonction pré-addeur/soustracteur pour l'opération $x-x0$ et d'effectuer la multiplication et l'addition finale par y0. Cette implémentation est représentée par la figure 3.5 représentant d'une part le schéma du calcul et d'autre part l'utilisation du DSP.

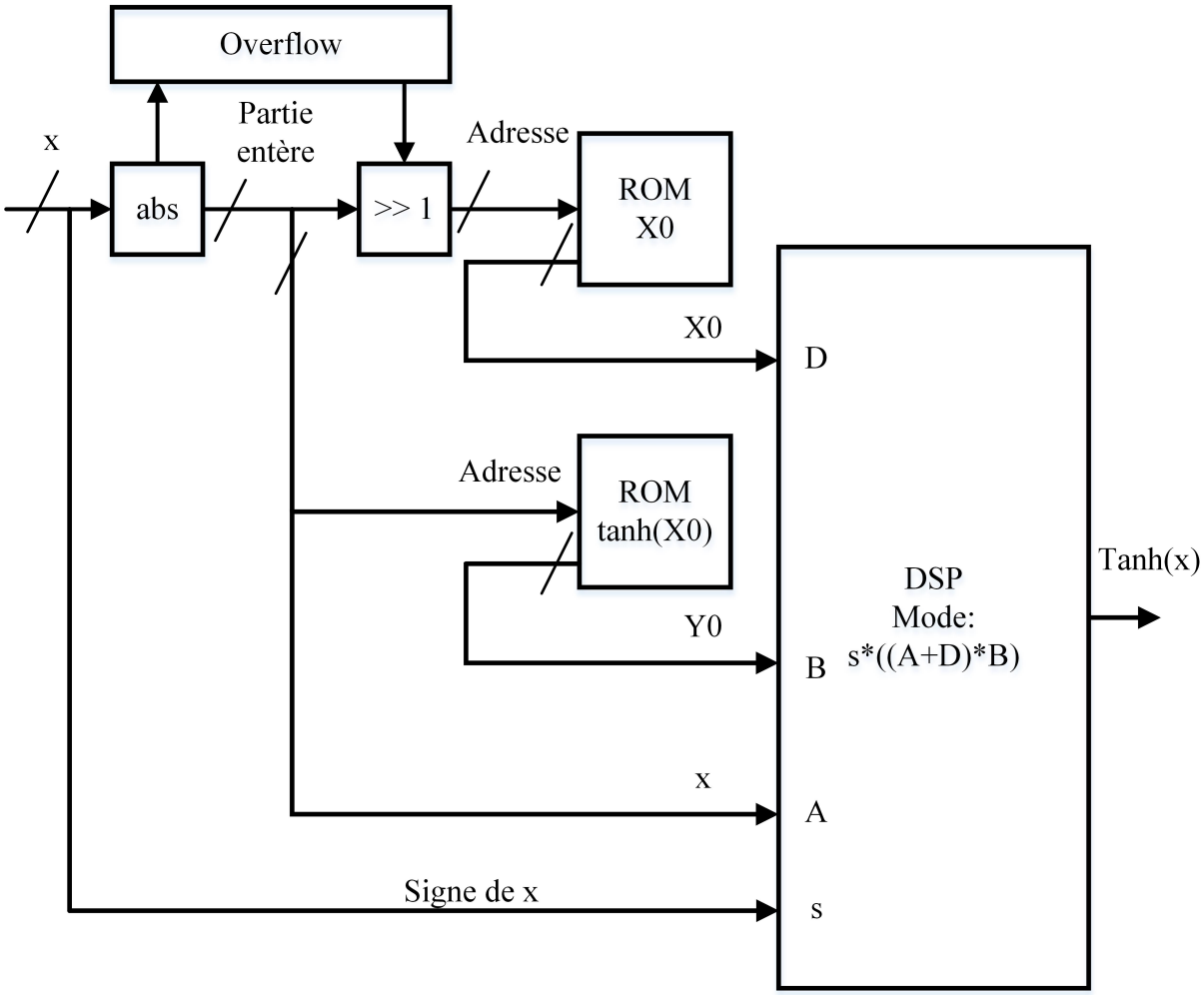

FIGURE 3.5 – Implémentation de l'interpolation de Lagrange : étant donné que la ROM de X0 contient des valeurs par pas de 0,5, un décalage de bits permet de sélectionner les valeurs de X0 correspondant aux valeurs de x le plus proche.

3.5.3 Comparatif

L'interpolation de Lagrange reproduit bien le fonctionnement attendu, cependant, les ressources nécessaires dépendent de la précision voulue. Le système actuel à base de CORDIC possède la meilleure précision pour le calcul des équations de Hodgkin-Huxley. Il est possible de réduire le nombre d'itérations afin de réduire les temps de propagation et le nombre de LUT utilisés. La figure 3.6 montre l'erreur absolue du calcul du CORDIC en fonction de l'angle et du nombre d'itérations. On peut observer qu'au-delà de 10 itérations le calcul converge vers la solution avec une erreur minimale.

	CORDIC	Interpolation
Cycles	10	3
Fréquence max (MHz)	100	500
Précision	2,673.10-4	5,5.10-2
LUT	700	96
FF	214	24
DSP	1	1

TABLE 3.1 – Comparaison des performances du CORDIC et de l'interpolation : la précision est l'erreur quadratique moyenne calculée avec un pas de 0,01 radian.

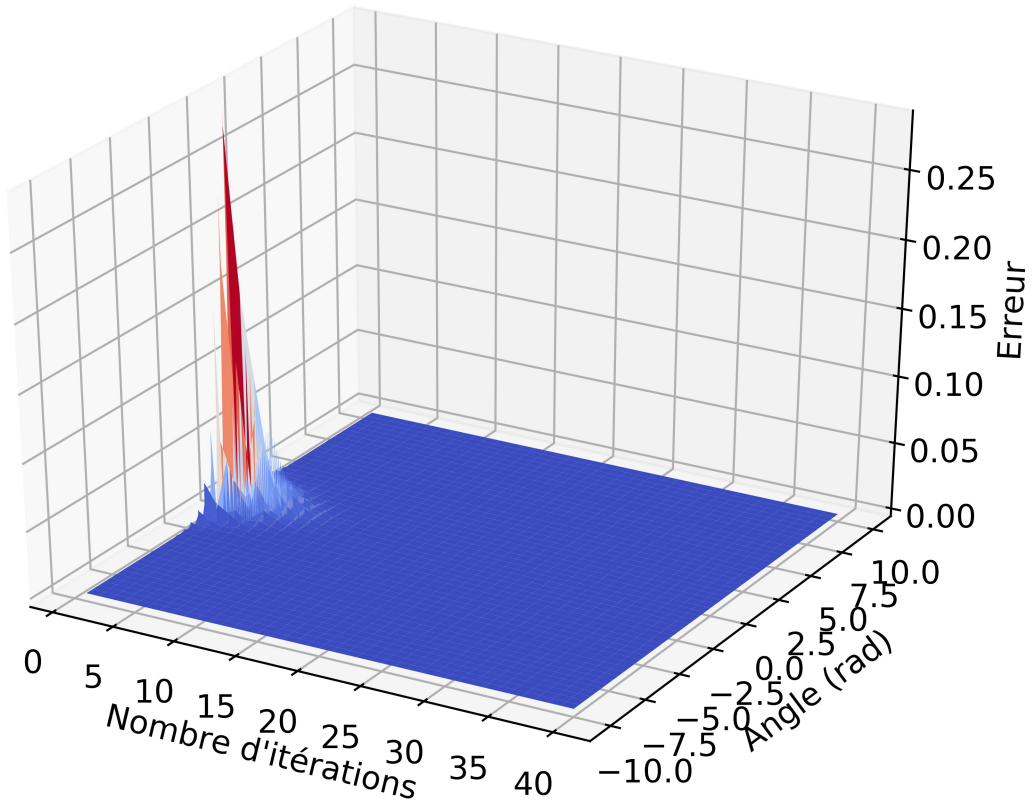

FIGURE 3.6 – Graphiques représentant la variation de la précision du cordic en fonction du nombre d'itérations.

La figure 3.1 montre que l'utilisation de l'interpolation possède plus d'avantages à l'implémentation, car elle est peu coûteuse en ressources et est éligible à de plus hautes fréquences. Cependant, le potentiel de l'interpolation se fait au prix d'une précision moins importante. Or, les réseaux de neurones nécessitent une précision relativement importante et posséderont, de ce fait, des modules de calcul à base de CORDIC. Le module de calcul par interpolation peut être utilisé dans des applications qui nécessitent moins de précision. Il est aussi possible d'améliorer la précision de l'interpolation en utilisant plus de points ou en augmentant la taille des vecteurs pour l'encodage des données.

3.6 Résolution d'équations différentielles

Cette section a pour but d'énoncer plusieurs méthodes et de justifier le choix qui a été fait. Toutes les méthodes ne seront pas citées au vu du nombre d'algorithmes. Les choix d'implémentation se font principalement sur un compromis entre la complexité des méthodes, comprenant le coût d'implémentation et le temps de calcul, et la précision de ces méthodes. Pour commencer, les systèmes numériques sont des systèmes dits discrets, car ils possèdent un nombre fini d'états pour représenter un système. Ces états sont séparés par des « pas » et correspondent, dans un système dépendant du temps, au moment où les calculs commencent. Les équations différentielles sont calculées à chaque pas et nécessitent un solveur. Parmi ces derniers, il existe les solveurs à pas adaptatifs. L'intérêt d'une telle méthode se situe dans sa capacité à réduire la taille des pas lorsqu'il y a besoin de précision et, dans le cas contraire, à élargir le pas, ce qui a pour effet de réduire le temps d'exécution. Le problème de cette méthode est que, dans le cas d'une implémentation numérique d'un réseau de neurones, les neurones sont calculés en utilisant le même solveur dans un système de *pipelines*. Le fait d'avoir un pas différent pour chaque neurone rendrait les calculs plus difficiles. Pour les pas non adaptatifs, les algorithmes ODE de Matlab, l'algorithme d'Adams-Bashforth-Moulton Predictor-Corrector, Runge-Kuttas ou l'algorithme exponentiel-intégrateur sont des méthodes complexes et d'ordre supérieur à 1, rendant l'implémentation difficile avec un long temps de calcul. Une alternative simple et efficace est la méthode d'Euler. La méthode d'Euler se base sur la définition de la dérivée (équation 3.10). En modifiant l'équation 3.10, il est possible de produire l'équation 3.11 afin de retrouver le résultat de $f(x+\Delta_x)$ qui correspond à la valeur du prochain f(x). L'équation discrétisée est ainsi produite (équation 3.12).

$$\frac{df(x)}{dx} = \frac{f(x+\Delta x) - f(x)}{\Delta x} \tag{3.10}$$

$$f(x+\Delta x) = f(x) + \Delta x \frac{df(x)}{dx} \tag{3.11}$$

$$f_{n+1} = f_n + \Delta x \frac{df}{dx} \tag{3.12}$$

Au vu de la définition de la dérivée, on peut en déduire que plus le Δ_t approche 0 et plus la précision sera grande, ce qui a pu être montré par les simulations. Vu sous cet angle, il est donc impératif de trouver un pas de calcul suffisamment petit pour avoir une certaine précision mais aussi assez grand pour effectuer le calcul de plusieurs neurones et synapses entre deux pas. De manière à éviter l'utilisation de multiplications, le choix d'un pas sous forme d'une puissance de deux sera préférable afin d'effectuer un décalage de bits. Pour pouvoir choisir le pas idéal, des tests, dont les résultats sont présents sur la figure 10, ont été réalisés sur les équations de Hodgkin-Huxley, montrant la variabilité de la fréquence des *spikes* et de l'amplitude des potentiels d'action en fonction de plusieurs valeurs de pas de calcul. Les résultats montrent que l'augmentation de la taille des pas (dt) impacte le système en le rendant de plus en plus instable. D'une part la figure 10.a) montre que la fréquence diminue légèrement et progressivement jusqu'à tendre vers 0 (pas de potentiels d'action) lorsque le dt dépasse 0,08 ms. D'autre part, la figure 3.7.b) montre que l'amplitude des spikes affiche une variabilité de plus en plus grande au fur et à mesure que le pas augmente. Au vu de ces données, le choix optimal considéré est $dt = 2^{-5}$ et sera le pas par défaut utilisé dans la suite du manuscrit.

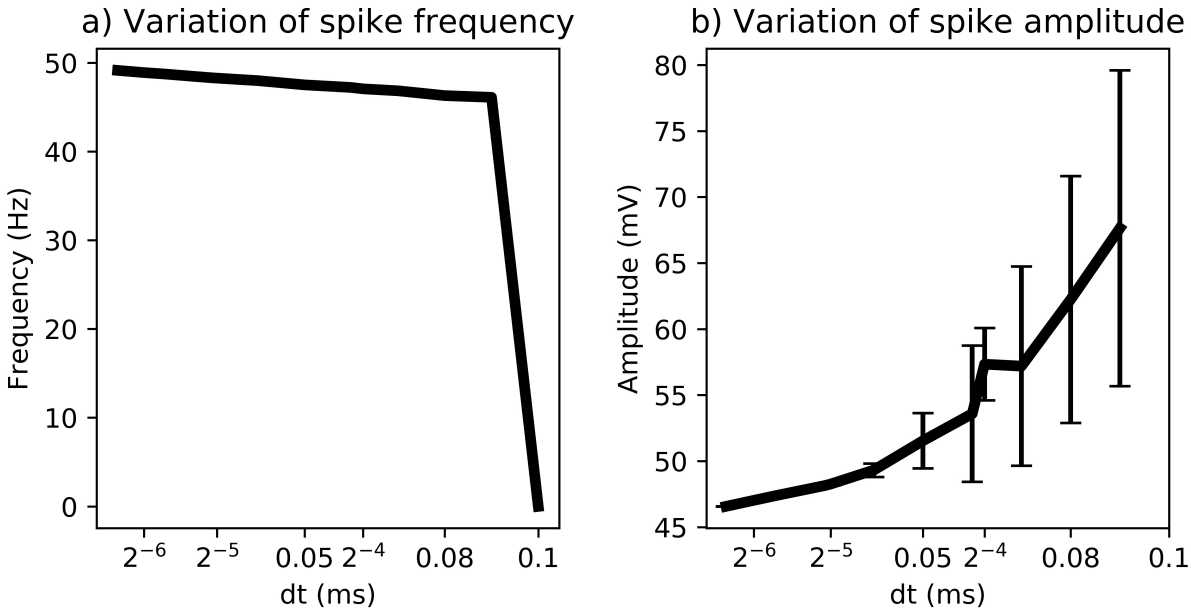

FIGURE 3.7 – Effet de la variation du pas sur les équations de Hodgkin-huxley : dt représente le pas de calcul, soit le temps séparant deux points sur les courbes : a) La fréquence de *spikes* varie en fonction de la valeur du pas choisi, une puissance de 2 permet de remplacer une multiplication par un décalage de bits ; b) L'amplitude des *spikes* varie en fonction des pas.

3.7 Produit matrice vecteur

Les matrices ont depuis longtemps suscité l'intérêt des mathématiciens et physiciens. Au fil des années, l'étude des différentes branches de la physique et des mathématiques a amené à construire des matrices plus grandes et plus complexes. Cette complexité grandissante s'accompagne d'opérateurs relativement complexes. Dans le précédent chapitre, il a été évoqué qu'il était possible de représenter aussi bien la connectivité synapses/soma et la connexion entre les dendrites, les axones et le soma dans la construction d'un modèle à compartiments par le biais d'une matrice. Cette représentation n'est pas anodine, car elle permet de simplifier les calculs. De plus, cette partie a pour but de montrer que non seulement il est plus simple d'utiliser des calculs matriciels sur FPGA mais qu'il est aussi possible d'atteindre de grandes performances.

3.7.1 État de l'art

Le nombre de données à traiter augmente d'année en année, ainsi grandit avec eux le besoin en performances en matière de calcul pour le traitement de ces données. La complexité grandissante liée à nos avancées dans la compréhension de la physique amène à développer des méthodes et algorithmes permettant d'exécuter des calculs matriciels de plus en plus grands. Avant d'établir l'état de l'art des algorithmes et implémentation FPGA des produits matrice vecteur, il est important de définir les termes.

Produit matrice vecteur

Le produit d'une matrice par un vecteur fait partie des opérations matricielles. Elle consiste, lorsque certaines conditions sont respectées, par une multiplication de chaque élément du vecteur par chaque élément des lignes de la matrice. Pour qu'un produit matrice vecteur puisse s'opérer, il faut que la taille d'une ligne de la matrice soit égale à la taille de la colonne du vecteur. Prenons une matrice M de taille $N \times N$ et un vecteur V de taille $1 \times N$, le résultat d'un produit de M par V sera un vecteur X de taille $1 \times N$ (équation. 3.13). Le calcul peut être décrit mathématiquement par l'équation 3.14.

$$A_{m,n} = \begin{pmatrix} M_{1,1} & M_{1,2} & \cdots & M_{1,n} \\ M_{2,1} & M_{2,2} & \cdots & M_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ M_{m,1} & M_{m,2} & \cdots & M_{m,n} \end{pmatrix} \cdot \begin{bmatrix} V_1 \\ \vdots \\ V_N \end{bmatrix} \tag{3.13}$$

$$X_i = \sum_j w_{ij} V_j = w_{i1} V_1 + w_{i2} V_2 + \cdots + w_{iN} V_N \tag{3.14}$$

Il existe plusieurs propriétés de matrices qui peuvent être exploitées et qui ont été exploitées via divers algorithmes et méthodes d'implémentation matériels qui sont présentés dans la partie suivante.

Algorithmes

Le besoin en temps et en taille pour le calcul matriciel a mené à développer des algorithmes tels que le *Kahan summation algorithm*, l'algorithme de Mailman, les algorithmes itératifs, le *Divide and conquer algorithm*, le *Sub-cubic algorithm*, les algorithmes parallèles et distribués. Cependant, ces algorithmes sont non seulement adaptés au calcul sur processeur mais en plus, pour beaucoup de ceux cités, ne concernent principalement que les produits de matrices par une autre matrice.

Implémentation matérielle

Concernant les méthodes de calcul de produit matrice vecteur sur *hardware*, la problématique est étudiée depuis longtemps en commençant par une implémentation sur puce de matrices systoliques [KUNG 1978]. Basées sur le même principe, plusieurs implémentations ont suivi, comme les puces permettant de calculer un produit matrice vecteur en utilisant la propriété des matrices à bande [MELH 1988]. Le coût et la complexité des systèmes analogiques ont mené aux implémentations numériques présentées par exemple via les travaux de [DELO 2005] et [PRAS 2007]. Et, plus récemment, [HU 2016] propose l'utilisation des memristors pour réaliser les produits matrice vecteur. Le problème est que la plupart des implémentations présentées sont des matrices *sparses* ou à bandes. De plus, dans le cas du memristor, la technologie actuelle ne permet pas de créer de grandes matrices d'autant plus qu'elles sont difficiles à reconfigurer.

3.7.2 Méthodes proposées

Plusieurs méthodes ont été précédemment proposées permettant chacune de répondre à des problématiques différentes. Les cas des calculs pour réseaux de neurones biomimétiques nécessitent une méthode de calcul peu coûteuse en ressources FPGA. De plus, la

matrice de connectivité sera possiblement dense et non symétrique. Enfin, le temps de calcul du produit matrice vecteur ne devra pas dépasser le pas déterminé pour le calcul de Hodgkin-Huxley, notamment pour la résolution des équations différentielles à travers la méthode d'Euler.

Contraintes matérielles

Au vu de la nature du produit matrice vecteur, on peut déjà déterminer que l'algorithme utilise des multiplications et des additions. En fonction du besoin, il est possible de créer une architecture qui utilise un nombre limité d'opérations en partageant les opérateurs et en rebouclant sur eux-mêmes. L'avantage de la méthode est qu'il est possible de réduire le nombre de ressources. Cependant, cette méthode prend du temps à s'exécuter, soit plusieurs cycles d'horloge. Une autre possibilité est d'utiliser tous les opérateurs en parallèle afin de réaliser la fonction souhaitée dans un temps bien plus court. Le problème de cette dernière méthode est qu'elle prend beaucoup de ressources. Selon la taille de la matrice, l'implémentation nécessitera un FPGA plus puissant et, si la taille est trop grande, l'implémentation pourrait même ne pas être possible. Un autre aspect des contraintes dues à l'implémentation de ce type d'opération vient du stockage et de l'accessibilité à la mémoire. Les FPGA modernes sont dotés de trois types de mémoire. É tant donné le besoin de pouvoir réécrire la mémoire à tout moment, les *Read Only Memory* (ROM) sont exclues. Aussi, au vu de la complexité de l'utilisation des DDR accessibles dans certaines cartes de développement, il a été décidé de ne pas les utiliser. Tout d'abord, il est possible de stocker des données dans les LUT. En effet, elles seront mise à jours à chaque cycle d'horloge par l'intermédiaire des bascules D. Ainsi, chaque donnée, chaque valeur peut être accessible simultanément sans contraintes particulières. Le problème est que les LUT sont une ressource limitée et que l'accessibilité simultanée à ces ressources peut créer des problèmes à l'algorithme de routage du constructeur pour interconnecter les composants. Certains LUT peuvent aussi fonctionner dans un autre mode, ces LUT sont appelés LUTRAM et constituent la mémoire dite distribuée du FPGA. Ce type de mémoire peut être contraignant, car elle a besoin d'un port d'écriture et d'un port de lecture. Cependant, la documentation de Xilinx pour les FPGA de la série 7 indique qu'il est possible de monter jusqu'à 8 ports dont un port d'écriture. Ce type de mémoire ne permet donc de lire que sept données à chaque cycle d'horloge et a besoin d'un cycle supplémentaire pour l'écriture des données. Enfin, le FPGA comporte un autre type de mémoire qui permet de stocker plus de données. Il s'agit des « blocs RAM » qui contraignent le système à n'utiliser qu'un seul port, voire deux, en utilisant la technologie de Xilinx pour la lecture ou l'écriture. De plus, cette catégorie de RAM a besoin d'un cycle d'horloge pour l'écriture de données et d'un autre pour la lecture. Un autre aspect de l'utilisation des différents types de mémoire est l'éligibilité à fonctionner à haute fréquence. Le fait de pouvoir faire fonctionner un système à haute fréquence permet d'exécuter plus de calculs et plus rapidement.

Calcul matriciel pour réseaux de neurones biomimétiques

La contrainte de temps accordée pour pouvoir calculer un produit matrice vecteur s'élève initialement à un pas de calcul. Cependant, il est possible d'utiliser des pas de calculs différents comme le montre [POLY 1964] pour la matrice et le reste du calcul. Des tests empiriques ont montré une détérioration de la précision et une instabilité du système lorsque le pas dépasse $dt = 2^{-4}$. Les tests ont été faîs en calculant les équations de Hodgkin-Huxley en utilisant les paramètres de [POSP 2011] et en utilisant les synapses de [DEST 1998]. Le

pas de calcul général est inchangé et reste à $dt = 2^{-5}$ et le produit matrice vecteur est mis à jour tous les deux pas. Si N est le nombre de neurones maximal que le système peut calculer (limité par le pas de calcul) alors, le temps de calcul est donc de $\frac{N}{clk} + p$ secondes avec clk, la fréquence du système en Hz et p le temps nécessaire pour parcourir le calcul pipeline et afficher le premier résultat. Le temps de calcul du produit matrice de connectivité et vecteur de tension de membrane sera de $\frac{2N}{clk} + p$ s. Le premier pas servira à multiplier les composantes du vecteur avec les composantes de la matrice, et le second pas à additionner chaque valeur d'une même ligne de la matrice. Cette méthode permet de laisser du temps au cœur de calcul pour effectuer ses opérations, ce qui réduit les contraintes et les ressources permettant finalement de calculer N^2 synapses maximum.

Contrainte de vitesse

Pour d'autres applications, il est important d'effectuer des calculs rapides. Un exemple de problème d'optimisation combinatoire dont les équations sont inspirées de calculs neuromorphiques sera évoqué dans le prochain chapitre. Les matrices de ce type de problème sont de tailles variables pouvant aller de 100 à 3 000. De plus, les valeurs de ces matrices ne prennent que les valeurs -1, 0 ou 1. Pour pouvoir créer le circuit, nous prendrons le cas le plus complexe, à savoir un problème de taille 3 000. Une telle taille de matrice suggère qu'il faudrait N^2 multiplications et $\frac{N^2}{2}$ additions avec N la taille de la matrice et le 2 représente le nombre d'additionneurs nécessaires. Il n'est donc pas possible d'implémenter ces opérations dans un FPGA du commerce. Le choix s'est donc dirigé sur une décomposition de la matrice sous forme de plusieurs matrices de taille 100, car il s'agit d'un multiple des tailles des matrices qui seront traitées d'une part et, d'autre part, il s'agit de la capacité maximale que peut utiliser le FPGA Kintex 7 ultrascale. En effet, dans le cadre du projet d'optimisation combinatoire, une autre carte de développement a été utilisée : le Kintex ultrascale KU040.

L'une des problématiques de ce type de calcul est la transmission de la mémoire. Les données à stocker sont nombreuses et l'utilisation de plusieurs LUT ou LUTRAM entraînent des temps de propagation et des problèmes de congestion lors du routage, empêchant ainsi le fonctionnement à plus haute fréquence. L'utilisation de block RAM pour stocker les données a donc été retenue. L'inconvénient des RAM est le nombre limité de ports d'accès en lecture ou en écriture. Pour éviter ceci, il est possible d'utiliser la méthode de la figure 3.8. En effet, le fait de dupliquer les RAM permet d'accéder en parallèle à plusieurs données.

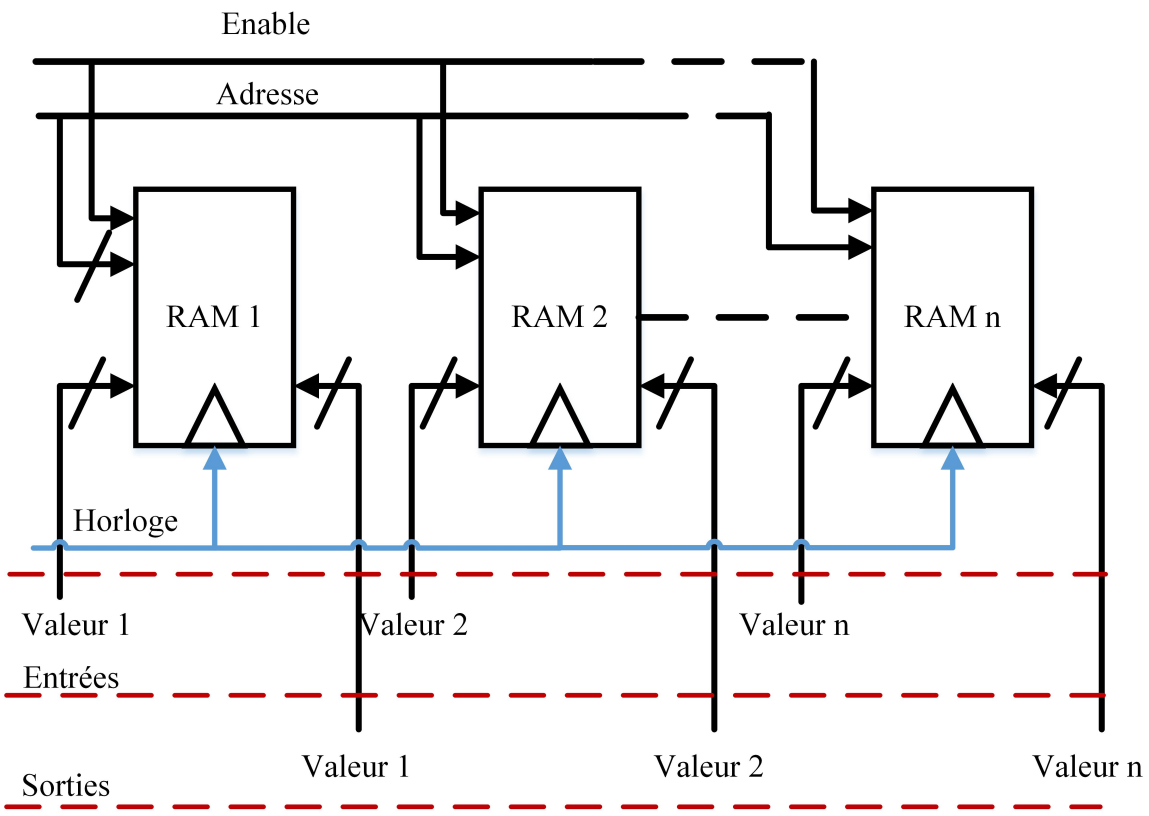

FIGURE 3.8 – Organisation des RAM de manière à accéder à 100 valeurs simultanément à chaque cycle d'horloge.

Les valeurs que peut prendre la matrice W sont m avec $m \in -1, 0, 1$ ce qui permet un codage en binaire sur 2 bits. Le vecteur X est codé sur 13 bits en virgule fixe avec 1 bit de signe et 12 bits de décimales. È tant donné que les valeurs ne dépassent pas 1, la partie entière a été ignorée afin de gagner des ressources. Il est donc possible de multiplier un vecteur de 2 bits par un autre de 13 bits ce qui correspond à une suite de portes logiques suivie d'une addition. En suivant cette méthode, il est possible de réaliser 1 000 multiplications suivies de 500 additions en un cycle d'horloge. En rebouclant 10 fois on obtiendrait le résultat du produit de W par X. Des tests ont été réalisés sur le FPGA montrant qu'il permettait à 50 MHz d'effectuer les additions de 300 valeurs en un cycle d'horloge. L'inconvénient d'une telle méthode est d'une part la création de longs temps de propagation et, d'autre part, le rebouclage empêchant le système d'être pipeline et de fonctionner à haute performance en matière de vitesse. Pour compenser cela, sachant que chaque bit des valeurs de sortie des blocs RAM passe par des LUT et des bascules D, il est possible de résoudre le problème par une équation logique préconfigurée dans les LUT à la place des multiplications. La figure 3.9 montre un tableau binaire représentant la sortie en fonction de W, une valeur codée sur 2 bits représentant −1, 0 ou 1. La figure montre aussi la représentation qu'il est possible d'avoir sous forme de LUT.

a)

W1	W0	S
0	0	0
0	1	P
1	0	--
1	1	$\bar{P}$

b)

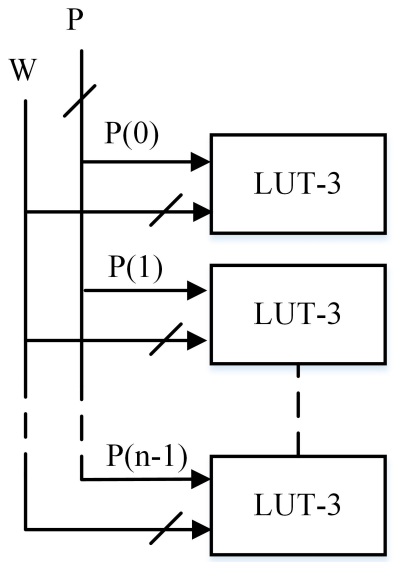

FIGURE 3.9 – Représentation sous forme de LUT à partir d'un tableau d'une multiplication de P par W; W est la représentation binaire d'une valeur de la matrice; P une valeur du vecteur à multiplier par la matrice; S le résultat de la multiplication de W par P; $\bar{P}$ représente le complément à 2 de P soit son changement de signe.

La représentation de la figure précédente ressemble tout simplement à un multiplexeur. L'équation logique du multiplexeur qui a été généré par le tableau binaire 3.9.a) est représentée par l'équation 3.15. Le terme $\bar{P}$ peut être trompeur, car il ne s'agit pas d'un « non » logique mais du complément à 2 de P. Ainsi, l'équation 3.16 remplace $\bar{P}$ par $\bar{P} \oplus C$.

$$P\bar{W}_1 W_0 + \bar{P} W_1 W_0 \tag{3.15}$$

$$P\bar{W}_1 W_0 + (\bar{P} \oplus C) W_1 W_0 \tag{3.16}$$

$$P\bar{W}_1 W_0 + \bar{P} W_1 W_0 \oplus C W_1 W_0 \tag{3.17}$$

$$W_0 (P \oplus W_1) \oplus C W_1 W_0 \tag{3.18}$$

$$W_0 (P \oplus W_1) \tag{3.19}$$

P est une représentation sur N bits des valeurs du vecteur avec $N = s + i + d$ sachant que s est le bit de signe, i le nombre de bits représentant la partie entière et d le nombre de bits représentant la partie décimale. C est la représentation sur N bits de 2^{-d} car le complément à 2 d'un vecteur binaire est l'inversion de chaque bit plus un « 1 » binaire et dans le cas d'une représentation d'un nombre binaire signé et sous forme de virgule fixe, plus 2^{-d}. L'équation 3.18 est générée en utilisant l'algèbre de boole et l'équation 3.19 en considérant d'accepter une imprécision lors de la simplification. Dans le cas où cette précision n'est pas acceptable, il est possible d'ajouter 2^{-N} au résultat de l'équation 3.19 en contrepartie de l'utilisation de ressources sous forme de LUT supplémentaires. L'approximation utilisée par l'équation logique 3.19 induit non seulement la propagation d'une erreur mais implique aussi l'utilisation d'un grand nombre de ressources qui peuvent rentrer dans le FPGA mais constituera, dans le cas d'un kintex 7, 60 à 70 % de ses ressources ce qui laissera peu de place pour d'autres opérations. Il est possible de réduire une nouvelle fois les ressources en optimisant leur utilisation. Si l'on reprend l'ordre des opérations d'un produit d'une matrice W par un vecteur X de taille 100 par 100, alors cela veut dire qu'il y aura 100^2 multiplications.

En utilisant les équations logiques précédentes, les ressources utilisées seront de $N \times 100^2$ LUT, avec N le nombre de bits représentant un vecteur. La première étape consiste à créer 100 vecteurs de $W_{ij}X_i$, contenant 100 valeurs, qui devront être additionnés entre eux. Il y a deux façons de réaliser une addition sur un FPGA Xilinx. La première est d'utiliser une équation logique et la deuxième est l'utilisation d'un composant présent dans chaque LUT. Ce composant est notamment présent dans les FPGA de type Ultrascale et est représenté par la figure 3.10.b. La particularité du CARRY8 est qu'il peut agir comme additionneur ou soustracteur. Les entrées et sorties sont encodées sur 8 bits et peuvent être routées en cascade de manière à additionner de plus grands vecteurs binaires. Il y a en plus un signal de 1 bit qui permet de sélectionner le mode additionneur ou le mode soustracteur. Le circuit généré dans le FPGA est représenté par la figure 3.10. Une valeur du vecteur sur deux est soumise à l'équation 3.19, les autres sont d'abord sélectionnées par un multiplexeur déterminé par le bit de poids faible de W avant de rejoindre l'additionneur dont le mode est contrôlé par le bit de poids fort de W. En voyant l'utilisation du mux dans le circuit, il serait légitime de se demander finalement pourquoi ne pas utiliser deux mux pilotés par les W pour chaque valeur de X. Cependant, pour de tels vecteurs, un grand nombre de mux en cascade est nécessaire impliquant alors plus de ressources au niveau des LUT.

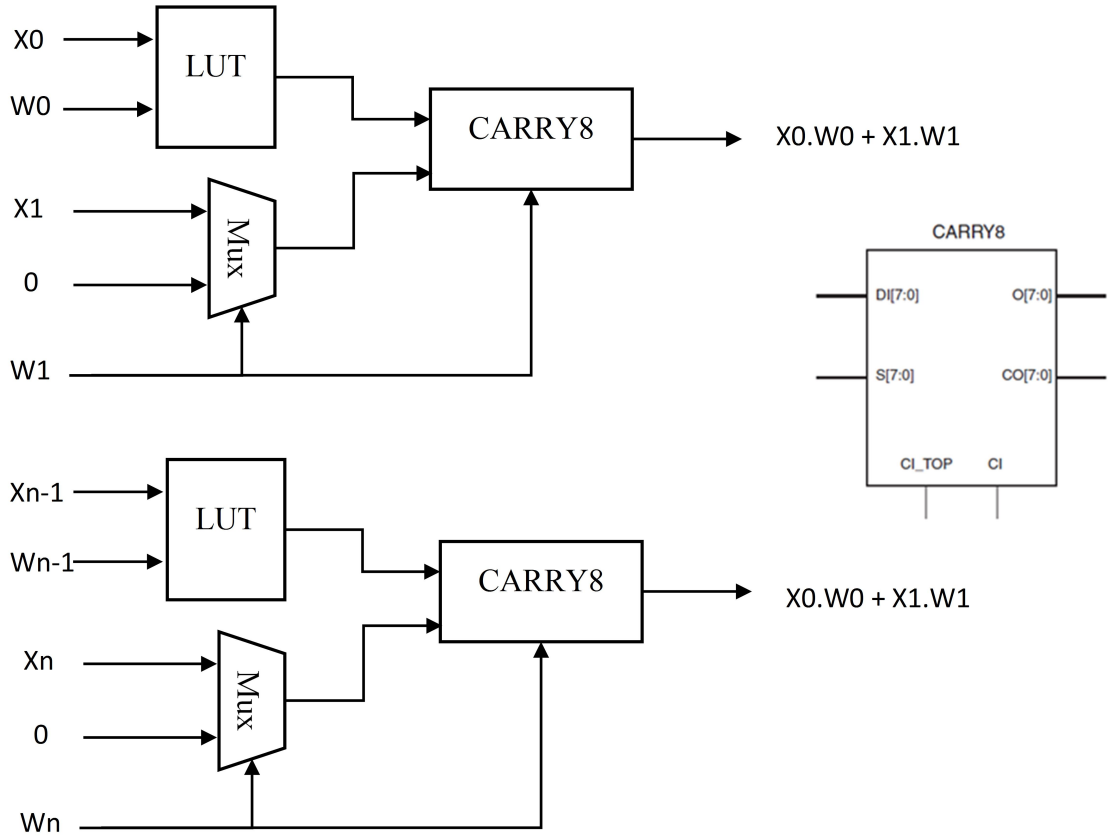

FIGURE 3.10 – Organisation des composants Carry8 des FPGA de type Ultrascale pour un produit des éléments d'une matrice par les éléments d'un vecteur.

La figure 3.10 montre la première étape du produit matrice vecteur et génère 100 vecteurs de 50 valeurs. En créant une autre étape d'additions de deux valeurs, il est alors possible de générer 100 vecteurs de 25 valeurs. Le reste des additions est exécuté par les DSP en mode addition et cascade afin d'obtenir les meilleures performances possible.

3.8 Protocoles de communication

Pour permettre une évolution et des modifications du réseau en temps réel pour les expériences biohybrides, une transmission d'informations rapide pendant la simulation est nécessaire. Il est alors possible de changer les paramètres biophysiques des équations de HH, de changer le courant de stimulation ou la matrice de connectivité par une interface sur ordinateur. Le but étant de pouvoir effectuer toutes les modifications le plus rapidement et simplement possible en cours d'expérimentation.

3.8.1 UART

Les choix de transmissions se sont portés sur l'*Universal Asynchronous Receiver Transmitter* (UART). Ce protocole de communication n'est pas le plus rapide, ni le plus efficace, cependant il a l'avantage d'être très simple à mettre en place. De plus, l'UART avait déjà été implémenté lors des travaux précédents de l'équipe. L'UART est un mode transmission série bidirectionnel (*full-duplex*) asynchrone. La vitesse de transmission peut être modifiée et l'information est composée par paquets de trames. Ces trames sont composées d'un bit de *start*, de 8 bits de données, d'un bit optionnel permettant la détection d'erreur et d'un bit de stop. Le FPGA possède une puce dédiée UART permettant d'effectuer des communications avec d'autres interfaces. Dans le cadre des travaux de thèse, ce mode de transmission sert à communiquer avec un ordinateur afin de modifier en cours de manipulation et en temps réel les paramètres biophysiques des neurones implémentés dans le FPGA et la matrice de connectivité servant à décrire les connexions synaptiques entre les neurones. Le système de transmission permet aussi de modifier le courant de stimulation appliqué aux neurones.

3.8.2 Visualiser les potentiels d'action avec un CNA

Certaines expériences biohybrides ou simulation de réseaux requièrent de pouvoir accéder au potentiel d'action sous sa forme analogique (par opposition à sa forme numérique présente dans le FPGA). Le meilleur moyen d'accéder au signal analogique à partir de données numériques calculées et générées dans le FPGA est d'utiliser un Convertisseur Numérique/Analogique (CNA). La marque Digilent qui est à la base de la carte de développement Genesys 2 propose en vente des extensions appelées « PMOD » qui peuvent être directement branchées sur la carte. Le choix s'est porté sur deux d'entre elles, le PMOD DA4 qui propose un CNA AD5628 sur 12 bits avec 8 sorties. Le deuxième choix, s'il y a besoin de plus de précision, est le PMOD DA3 contenant un AD5541A sur 16 bits. Ces PMOD sont accessibles depuis le FPGA en utilisant une communication via le mode de transmission *Serial Peripheral Interface* (SPI).

Le SPI, est un mode de transmission bidirectionnel (*full-duplex*) série et synchrone permettant de communiquer avec plusieurs machines. Il est composé de 4 connexions minimales permettant d'assurer la transmission d'un maître (machine principale) à plusieurs esclaves. Le maître se connecte aux esclaves en transmettant une horloge sur une sortie et des informations via un autre pin MOSI (*Master Out Slave In*), aux machines esclaves. Il reçoit également des informations via un pin MISO (Master In Slave Out). Enfin, la machine maitresse sélectionne ses machines esclaves par un signal de sélection à l'état haut (niveau logique égal à 1) par défaut et passe à un état bas (à 0) lorsque la machine souhaite communiquer.

3.9 Architecture des réseaux de neurones

Toutes les fonctions qui ont été nécessaires au bon fonctionnement du réseau de neurones ont été détaillées dans ce chapitre. Ainsi, cette dernière partie sera consacrée à spécifiquement détailler l'architecture du réseau et les choix qui ont été faits afin d'optimiser le système neuromorphique.

3.9.1 Architecture générale

Dans les langages de description tels que le VHDL ou le verilog, les circuits sont hiérarchisés et forment un arbre dont le sommet est nommé le «*top level entity*». L'architecture générale du design se résume donc par la composition de ce top qui peut être retrouvé à travers la figure 3.11. Le module clock handle est un module de gestion de l'horloge. Il s'agit d'un circuit dédié composé d'une Boucle à verrouillage de phase ou PLL permettant de réduire ou d'augmenter la fréquence. Dans notre cas, le circuit sert à convertir d'une part une horloge sur un mode de transmission différentiel à un mode de transmission sur une ligne unique. D'autre part, il permettra de réduire la fréquence en passant de 200 MHz à 100 MHz. Les systèmes de transmission d'informations sont au nombre de deux. Le *Serial Peripheral Interface* (SPI) est un mode de transmission série qui sert ici à piloter le CNA afin d'obtenir des sorties analogiques reproduisant les potentiels d'action. L'UART permet de recevoir les paramètres de Hodgkin-Huxley, les valeurs de courant de stimulation, la matrice de connectivité et les conductances des synapses. Il permet aussi la transmission de bits servant à reproduire un *rasterplot*. Pour rappel, le *rasterplot* est un graphique montrant les moments où les neurones « spikent » en fonction du temps. Ces bits sont gérés par un autre module permettant de mettre à un les bits correspondants aux neurones dont le potentiel d'action a dépassé le seuil de 0 V. Enfin, les deux modules restants sont le processus de calcul principal et un gestionnaire de RAM. Ce dernier, active et envoie les adresses aux RAM afin que l'information se propage dans les circuits de calcul.

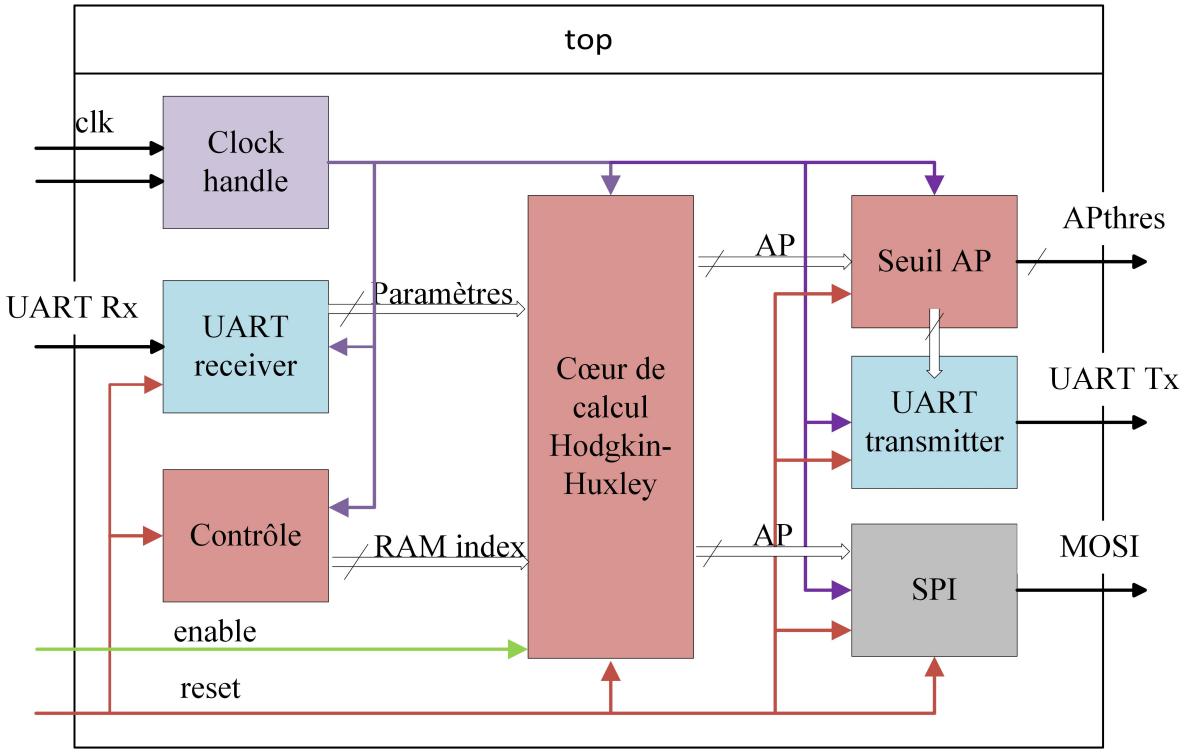

FIGURE 3.11 – Architecture générale du réseau de neurone. L'horloge est une horloge différentielle, elle est gérée par une PLL (violet) pour réduire la fréquence et générer une ligne unique de transmission. Les paramètres de Hodgkin-Huxley et de connectivité des synapses sont reçus par UART (Bleu) et sont envoyés dans le cœur de calcul (rouge) ; Un module de contrôle permet de gérer les index des RAMs et des neurones à calculer (rouge) ; Un module appelé « seuil AP » permet de détecter le seuil d'un potentiel d'action et de générer un « 1 » logique lorsqu'un neurone spike (rouge). Cette information est externalisée en dehors du FPGA et est envoyée via UART ; Le dernier module permet d'envoyer les potentiels d'action via SPI (gris).

3.9.2 Architecture spécifique

L'architecture générale du système est définie par deux modules. Il y a d'abord le gestionnaire du réseau qui contrôle les moments de lecture et d'écriture des RAM ainsi que le module de calcul. Ce dernier présente une architecture assez particulière, représentée par la figure 3.12, car elle est *semi-pipeline*. Le module de calcul des fonctions de canaux ioniques est piloté par une machine d'état permettant de sélectionner les coefficients à utiliser avec les équations simplifiées. Les constantes sont envoyées simultanément au module de calcul de x_{inf}, contenant le calcul de la tangente hyperbolique, et au module de calcul τ_x contenant le cosinus hyperbolique. Les résultats des deux modules mentionnés se rejoignent pour décrire, en fonction de la tension de membrane reçue, une fonction d'activation ou d'inactivation des canaux ioniques. Ce système reboucle autant de fois qu'il y a de canaux ioniques à calculer. Cette méthode de calcul permet de limiter l'utilisation des ressources et malgré tout de garder un nombre de neurones supérieur à ce que la littérature offre (voir Chapitre 4). En parallèle du calcul des canaux ioniques, le module de calcul de courant ionique et de tension de membrane s'effectue en utilisant les anciennes valeurs de canaux ioniques. La tension de membrane est calculée avec les $n^{\text{èmes}}$ valeurs des canaux ioniques pendant qu'un autre module calcule les n+1.

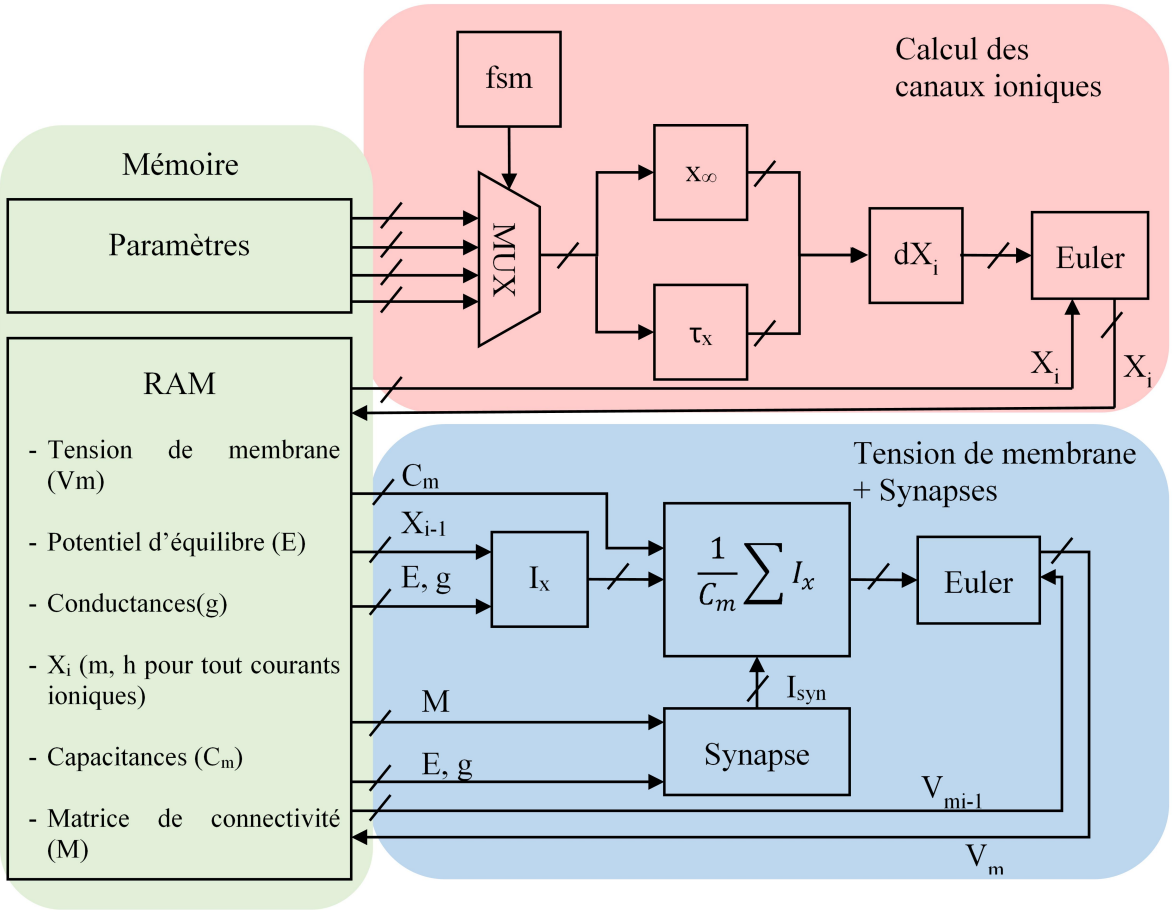

FIGURE 3.12 – Architecture du réseau de neurones.

Le système cité précédemment peut être décrit par des chronogrammes (figure 3.13). Á chaque cycle d'horloge correspond une information relative au type de canaux ioniques calculés. Le chronogramme en figure 3.13.a) montre la propagation des informations correspondant aux courants ioniques pour un seul neurone et 6 courants ioniques. Les lettres représentent des paramètres constants qui sont utilisés par les équations simplifiées. La figure 16.b) montre comment les informations, présentes sur le chronogramme, se propagent à travers les *pipelines* de calcul. Elle montre aussi le nombre de cycles d'horloge qu'il faut pour que l'information se propage sur l'ensemble des pipelines. Á 50 MHz il faut seulement 8 cycles d'horloge pour calculer la valeur d'un canal ionique. Le facteur limitant l'augmentation de la fréquence est le module de calcul de la division qu'il faut doubler pour pouvoir fonctionner à 100 MHz. De ce fait, le calcul passe à 9 cycles.

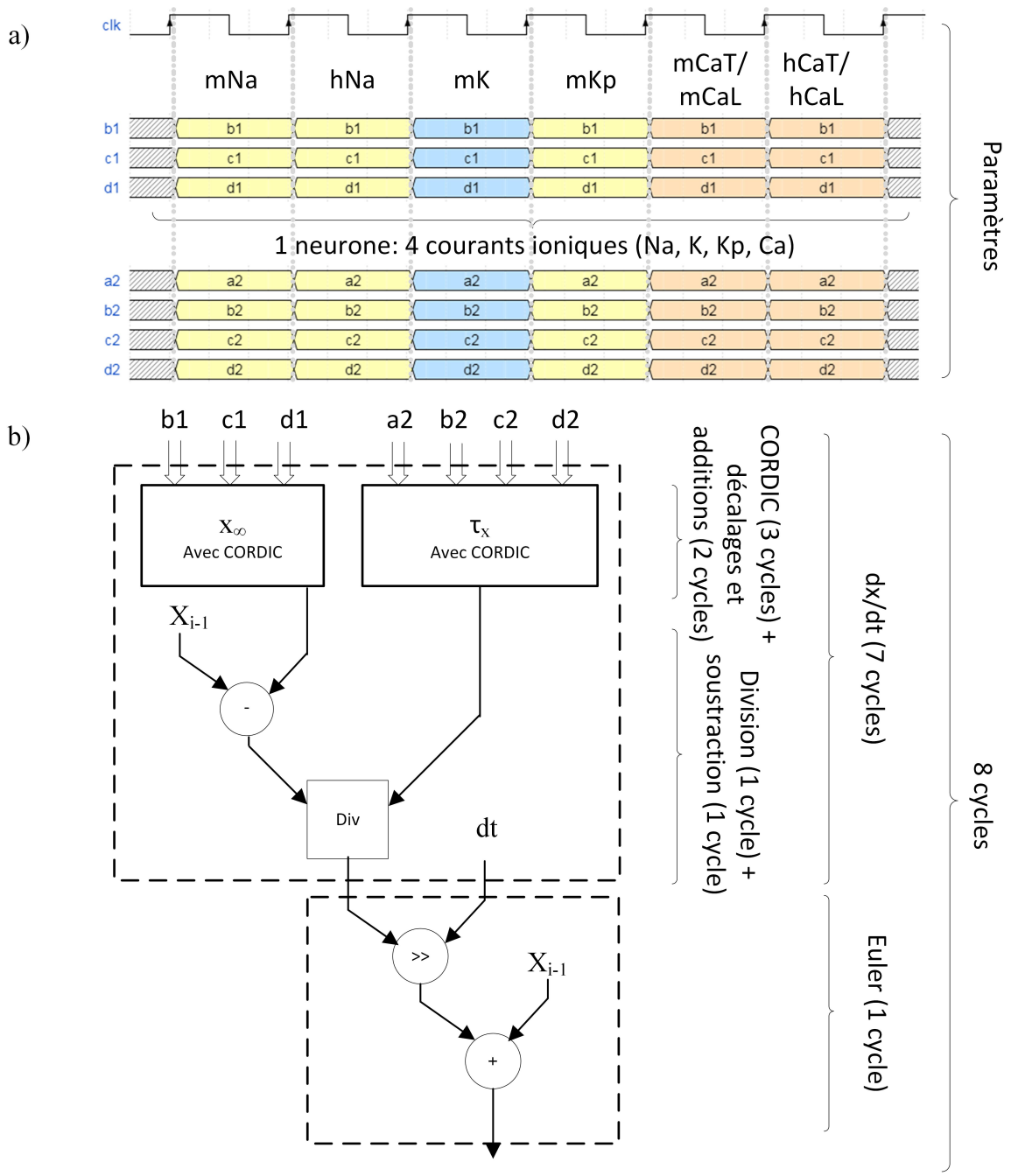

FIGURE 3.13 – Timing et architecture du calcul d'un canal ionique à 50 MHz.

3.9.3 Ressources

L'efficacité de l'architecture FPGA peut se résumer à sa fréquence de fonctionnement, le nombre de neurones et de synapses qu'il peut simuler et aussi par le nombre de ressources utilisées. Bien sûr, selon le FPGA, le nombre de ressources peut varier, cependant, il se composera toujours de LUT, de Flip-Flops (FF ou bascule D) et de DSP. Ces données ont été résumées dans le tableau 3.2 et sont comparées aux systèmes de l'état de l'art. Á noter que les implémentations FPGA que la littérature propose ne sont dotées que d'un seul type de neurone, à savoir les neurones du type Fast Spiking (FS). Or, l'implémentation proposée possède 4 types de neurones qui correspondent aux différents types de neurones corticaux.

Elle inclut également le bruit synaptique et N2 synapses pouvant être paramétrées pour être excitatrices ou inhibitrices. De plus, le système présent permet une modification en temps réel des paramètres biophysiques de Hodgkin-Huxley et de la connectivité des neurones. Ce système est aussi comparé à une implémentation des équations de Hodgkin-Huxley avec les paramètres de Pospischil sans optimisation et sans modification.

	Etat de l'art		Travaux de thèse			
	[BONA 2014]	4 cœurs (FS) [AKBA 2019]	Hodgkin-Huxley sans optimisation (FS, RS, IB, LTS)	Un coeur de calcul (FS)	Un cœur (FS, RS, IB, LTS) + N^2 synapses + bruit	30 cœurs (FS, RS, IB, LTS) + N^2 synapses + bruit
N neurones	150	2.048	150	1.034	500	15.000
Fréquence (Mhz)	63,386	58,8	100	100	100	100
LUT	86.032	46.045	169.003	4.735	5.551	167.346
Bascule D	30.528	4.606	65.059	1.552	2.360	71.608
DSP	1.112	280	332	16	28	840

TABLE 3.2 – Tableau comparatif des différentes solutions et de l'état de l'art en fonction des ressources, du nombre de neurones et de synapses et de la fréquence maximale de fonctionnement.

3.10 Résumé

Ce Chapitre a eu pour rôle de présenter les spécificités techniques de l'implémentation numérique. De plus, plusieurs problématiques sont posées par les implémentations matérielles qui ont aussi été répondues dans ce chapitre. Tout d'abord, l'encodage des données est un choix primordial car il déterminera les algorithmes qui seront utilisés, les architectures qui seront mises en place et les outils qui seront utilisés. Ensuite, des notions importantes concernant les temps de propagation ont été abordées. Elles concernent les implémentations numériques et spécifiquement le fonctionnement des bascules D. Puis, plusieurs algorithmes et méthodes d'implémentation d'opérateurs mathématiques tels que les multiplications, divisions, tangentes hyperboliques, exponentielles et produits matrices vecteurs ont été détaillés. Le chapitre 2 a pointé la possibilité de traiter les calculs de synapses comme des produits de matrices par vecteurs, ainsi, des méthodes ont été développées et expliquées dans ce chapitre. Ces opérateurs sont essentiels afin d'effectuer les calculs requis aux équations de Hodgkin-Huxley. La configuration des paramètres biophysiques se fait par transmission UART via un ordinateur. Il est aussi possible de recevoir des données afin de produire un graphique présentant les moments où les neurones génèrent un potentiel d'action. Une autre possibilité via le protocole SPI est de passer par un CNA afin de visualiser le signal analogique des potentiels d'action. Enfin, tous les opérateurs et protocoles ont été rassemblés afin de former les systèmes de réseaux de neurones dont les équations ont été formulées dans le chapitre précédent. Ces réseaux de neurones ont servi dans le cadre de plusieurs projets. Ces projets seront décrits dans le prochain chapitre.

Chapitre 4

Application de la plateforme de calcul neuromorphique

4.1 Introduction

Le premier chapitre a permis d'introduire le sujet des réseaux de neurones et des expériences biohybrides. Les chapitres 2 et 3 ont détaillé le développement de réseaux de neurones artificiels et d'une bibliothèque VHDL permettant le développement de systèmes neuromorphiques. Ce dernier chapitre montre, d'une part, les résultats des réseaux de neurones conçus sur FPGA et, d'autre part, les applications de ces réseaux en simulation et dans les expériences biohybrides. Ce chapitre décrira aussi quelques systèmes dont la dynamique s'apparente à celles produites par les réseaux de neurones dans le but de montrer l'efficacité des méthodes proposées dans les chapitres précédents.

4.2 *Spiking Neural Network* : Validation

La plateforme de réseaux de neurones biomimétiques créée et présentée dans les précédents chapitres inclut la possibilité d'avoir plusieurs sorties analogiques des potentiels d'action des neurones sélectionnés. L'utilisation du PMOD DA4, contenant un Convertisseur Numérique/Analogique (CNA), permet d'avoir 8 sorties analogiques de neurones dont les paramètres biophysiques, les connexions synaptiques et le courant de stimulation sont paramétrables. Cette partie a pour but de succinctement montrer les résultats des implémentations sur FPGA et de valider l'efficacité des méthodes précédentes lorsqu'elles sont appliquées aux réseaux de neurones.

4.2.1 Neurones corticaux

Les neurones corticaux ont été implémentés sur FPGA en suivant les méthodes proposées par les chapitres précédents. La figure 4.1 montre 3 types de neurones mesurés à l'oscilloscope sur 3 sorties du CNA. La figure 4.1.a) représente un neurone de type *Fast Spiking* (FS), la figure 4.1.b, un neurone de type *Regular Spiking* (RS) et la figure 4.1.c, un neurone de type *Intrinsically Bursting* (IB). La spécificité de ces types de neurones est expliquée dans le chapitre 4.1. L'avantage de l'utilisation d'une telle plateforme est qu'il est possible de changer les paramètres biophysiques de chaque neurone en temps réel.

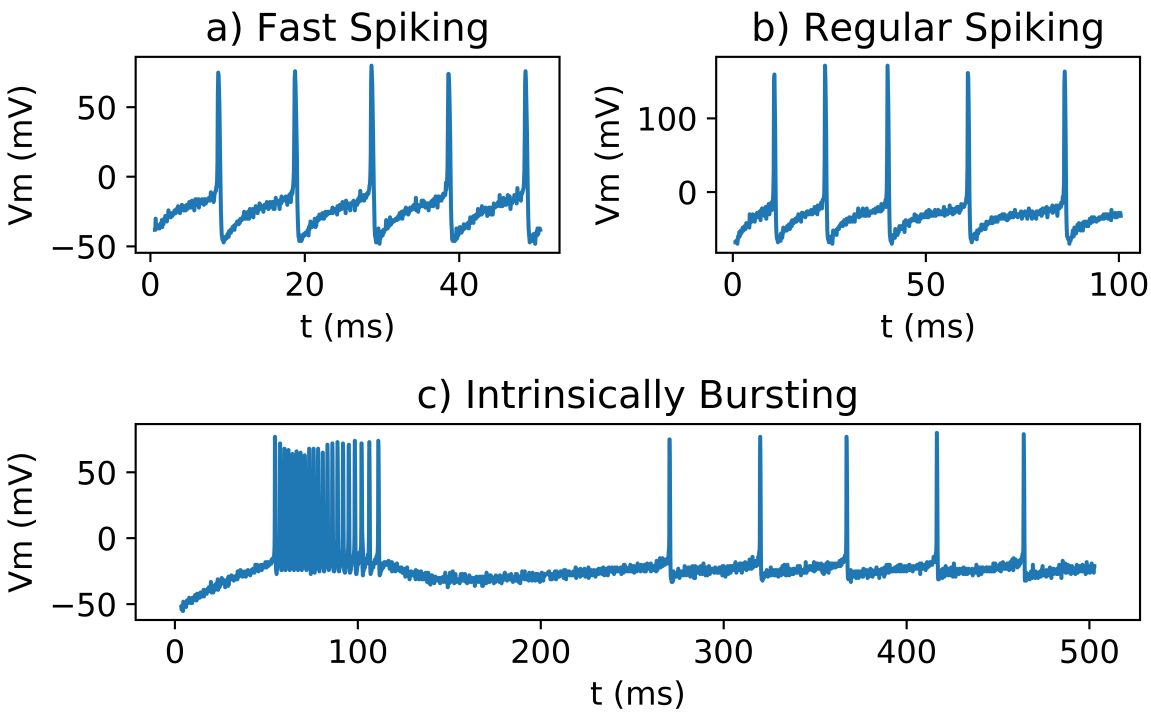

FIGURE 4.1 – Mesure à l'oscilloscope de 3 types de neurones : a) Neurone de type *Fast Spiking* (FS) ; b) Neurone de type *Regular Spiking* (RS) ; c) Neurone de type *Intrinsically Bursting* (IB).

4.2.2 Bruit

L'importance du bruit a déjà été expliquée dans le chapitre 1 et l'implémentation du modèle d'Ornstein-Uhlenbeck dans le chapitre 2. [NAWR 2007] a déterminé, à partir de plusieurs mesures chez le rat, que les cellules du néocortex possèdent en moyenne un *Inter-Spike Interval* (ISI) de 10,4 +/- 4,5 *spikes* par seconde. D'un autre côté, [WEBB 1976] a mesuré les activités spontanées des neurones de chat et publie les chiffres de 2,5 *spikes* par seconde. La figure 4.2 montre qu'il est possible de faire varier le taux d'ISI avec des paramètres différents sur les équations d'Ornstein-Ulhenbeck pour afficher un ISI proche de ce que [NAWR 2007] et [WEBB 1976] ont mesuré.

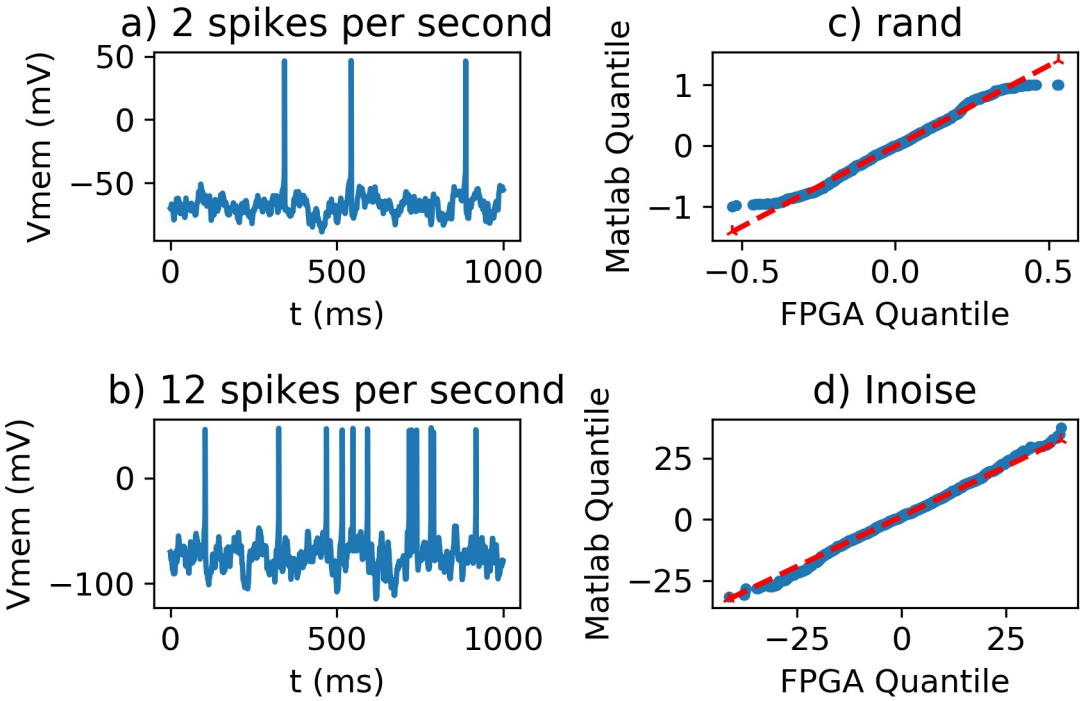

FIGURE 4.2 – Courant synaptique et activités spontanées appliqués sur deux neurones isolés : a) 2 *spikes* par seconde avec $\theta = 1$, $\mu = 0,1$, et $\sigma = 1,05$; b) 12 *spikes* par seconde avec $\theta = 1$, $\mu = 0.1$, et $\sigma = 2,2$.

4.2.3 Synapses

Le résultat de l'implémentation de la synapse peut être représenté, comme sur la figure 4.3.e), sous forme d'un graphique affichant la présence de potentiels d'action en fonction du neurone déclencheur. Les neurones sous la forme d'un point lorsque l'amplitude du PA dépasse les −10 mV (potentiel de repos autour de −60 mV). Ce type de graphique est nommé *rasterplot*. La figure 4.3.e) représente un *rasterplot* de 16 neurones montrant le comportement des récepteurs AMPA, NMDA, GABAa et GABAb. Les 16 neurones sont séparés par groupes de 4 neurones connectés. Les synapses d'un groupe sont les mêmes mais diffèrent d'un groupe à l'autre. Les neurones sont connectés en chaîne du neurone dont l'index est le plus faible au neurone dont l'index est le plus élevé.

AMPA - La figure 4.3.a) représente un neurone 1 connecté à 2 qui est à son tour connecté à 3. La fréquence du neurone 3 n'est pas assez forte pour stimuler le neurone 4, seule une faible oscillation provoquée par le courant synaptique est visible.

NMDA - Les neurones 5 à 8 possèdent des synapses NMDA. Ils fonctionnent comme la synapse AMPA mais mettent plus de temps à s'exciter. La figure 4.3.b) permet d'observer que les neurones mettent plus de temps pour exciter le neurone post-synaptique que dans le cas du récepteur AMPA.

GABAa - Les neurones 9 à 16 sont connectés par des synapses GABAa et sont constamment stimulés par un courant continu. Ils possèdent des synapses inhibitrices (Figure 4.3.c). Le

neurone 9 inhibe le neurone 10 qui ne peut plus inhiber le neurone 11.

GABAb - Dans le cas des neurones 13 à 16 (Figure 4.3.d), les synapses de type GABAb montrent une inhibition plus lente par rapport à la synapse GABAa. Ainsi, la figure 4.3 montre que le comportement des synapses implémentées correspond au résultat attendu.

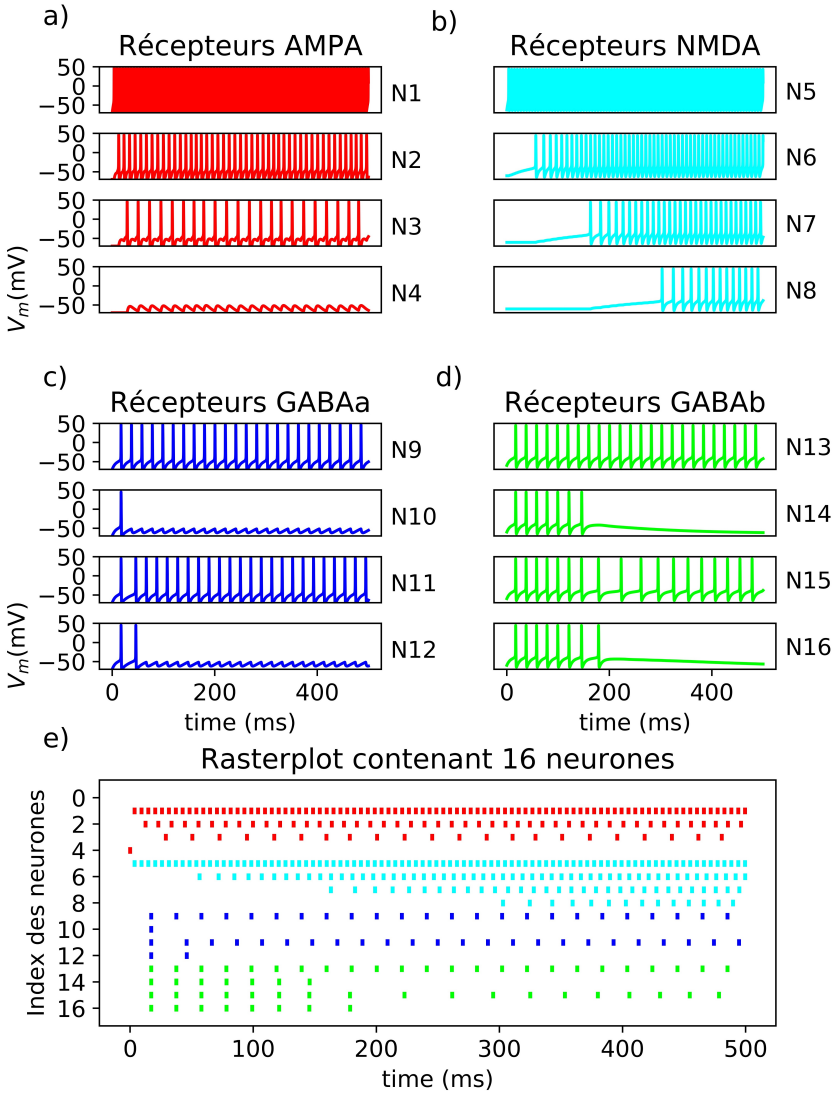

FIGURE 4.3 – Connexion synaptique montrant l'effet des différents récepteurs. Chaque neurone Ni est connecté et communique de manière unidirectionnelle avec Ni+1. Chaque récepteurs est indépendant des autres récepteurs : a) N4 est excité par N3 excité par N2 qui est à son tour excité par N1. L'excitation provoquée par N1 n'est pas assez forte pour générer un potentiel d'action chez N4 sur lequel on peut apercevoir les PSP ; b) Les récepteurs NMDA mettent plus de temps à être excités. Ainsi, un PA mettra plus de temps à survenir par rapport au cas du récepteur AMPA ; c) Les synapses de type GABAa sont inhibitrices. Le neurone 9 inhibe le neurone 10 et le neurone 11 inhibe le neurone 12 ; d) Les synapses de type GABAb mettent plus de temps à inhiber les neurones en comparaison des synapses GABAa.

4.2.4 Comportement de réseaux

Certains troubles neurologiques peuvent survenir par des lésions de la moelle épinière (voir chapitre 1). Des études ont montré qu'il était possible de rétablir artificiellement une connexion entre les parties affectées par le biais d'une stimulation électrique des nerfs [JACK 2016] [JOUC 2016]. La moelle épinière contient plusieurs réseaux de neurones responsables des mouvements moteurs par oscillation des neurones [KJAE 1996]. Ces réseaux se nomment *Central Pattern Generator* (CPG).

Les CPG sont des réseaux de neurones qui produisent un pattern de potentiels d'action (Figure 4.4) utilisés dans les fonctions motrices rythmées et répétitives telles que la marche, la respiration, la nage ou le vol, lorsqu'il n'y a pas d'informations reportées par des capteurs sensoriels [MARD 2001]. Le principe est basé sur la connexion entre deux neurones qui vont s'inhiber entre eux afin de former une oscillation. La figure 4.4.a) montre le schéma d'un réseau de neurones réalisant cette fonction d'oscillation qu'est le CPG et la figure 4.4.b) montre une simulation des équations proposées par [HILL 2001] basées sur des enregistrements biologiques des potentiels d'action d'un «*elemental oscillator*».

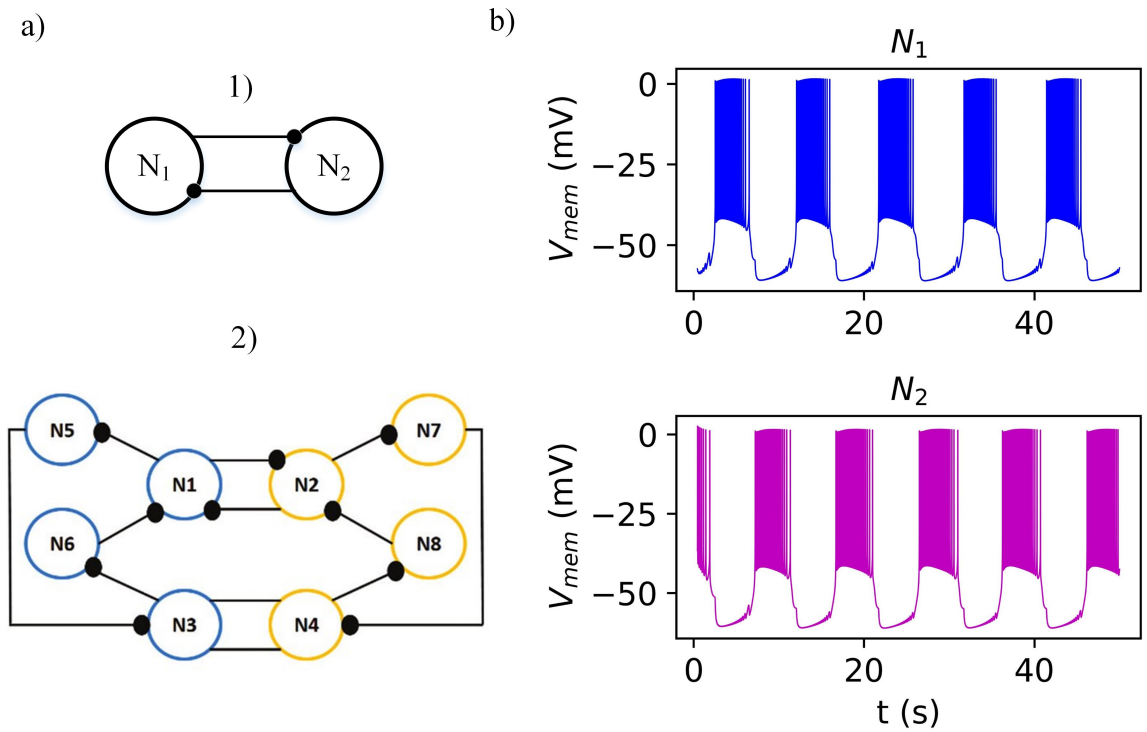

FIGURE 4.4 – Principe du CPG : a.1) Un «*elemental oscillator*» comme décrit par [HILL 2001] représentant deux neurones, N1 et N2, connectés entre eux par des synapses inhibitrices ; a.2) Cas d'un réseau de neurones plus complexe permettant de réaliser un CPG [BLAN 2019] ; b) Potentiels d'action générés par N1 et N2 représentés par a). Il est possible d'observer une oscillation sur différentes phases entre les deux neurones.

Par le passé, des réseaux de CPG ont été réalisés au sein de l'équipe montrant des résultats similaires à la biologie [AMBR 2013]. Le modèle de CPG est le modèle de [HILL 2001]. Ce dernier a réalisé, sur la base des équations de Hodgkin-Huxley, un modèle simulant les influx nerveux responsables du battement de cœur de sangsues. Le modèle de synapse utilise

l'équation 4.1 qui a été adaptée et implémentée avec les équations de HH.

$$I_{synS}(t,V) = (V_m(t) - E_{syn}) \sum_{i=1}^{N} \sum_{s=1}^{\infty} M_i \bar{g}_{synSi} f_{synSi}(t - t_s) \tag{4.1}$$

avec, I_{SynS} le courant synaptique, E_{syn} le potentiel d'équilibre représentant les ions de la synapse, $\bar{g}_{SynSi}$ la conductance maximale de la synapse, M_i la variable d'activation ou d'inactivation (représentée par les équations 4.2 et 4.3) et f_{SynSi} une fonction représentant la plasticité synaptique à court terme. Cette dernière est représentée par une exponentielle croissante et décroissante. Cependant [SIMO 2007] a montré qu'il n'était pas nécessaire d'utiliser cette fonction afin de réaliser un CPG. De ce fait, elle sera négligée lors de l'implémentation.

$$\frac{dM}{dt} = \frac{M_{\infty(V_pre)-M}}{0.2} \tag{4.2}$$

$$M_\infty = 0.1 + \frac{0.9}{1 + e^{-1000(V_pre+0.04)}} \tag{4.3}$$

En utilisant les méthodes présentées dans le chapitre précédent ainsi que les fonctions de la bibliothèque VHDL, il est possible de réaliser cette implémentation et de produire les résultats de la figure 4.5 qui représente un CPG généré par les équations de Hodgkin-Huxley et les synapses de [HILL 2001]. L'avantage de pouvoir modifier les paramètres biophysiques des modèles en temps réel est qu'il est possible de changer les fréquences et durées des CPG pour les adapter aux expériences. Au vu du nombre peu élevé de ressources requises pour faire fonctionner le réseau de neurones présenté dans ce manuscrit, il est possible d'utiliser des plateformes bien plus petites comme le CMOD A7 de la marque Digilent ce qui peut favoriser les expériences bio-hybrides et réduire la consommation électrique. Les ressources prises par le réseau de neurones sur le FPGA sont représentées sur le tableau 4.1.

	Utilisation	Disponible	Utilisation (%)
LUT	7257	20800	34.89
LUTRAM	630	9600	6.56
Bascule D	3662	41600	8.80
DSP	39	90	43.33

TABLE 4.1 – Ressources utilisées pour la génération de CPGs via le modèle de synapses proposé par [HILL 2001] combiné avec l'implémentation de Hodgkin-Huxley optimisée et proposée dans ce manuscrit.

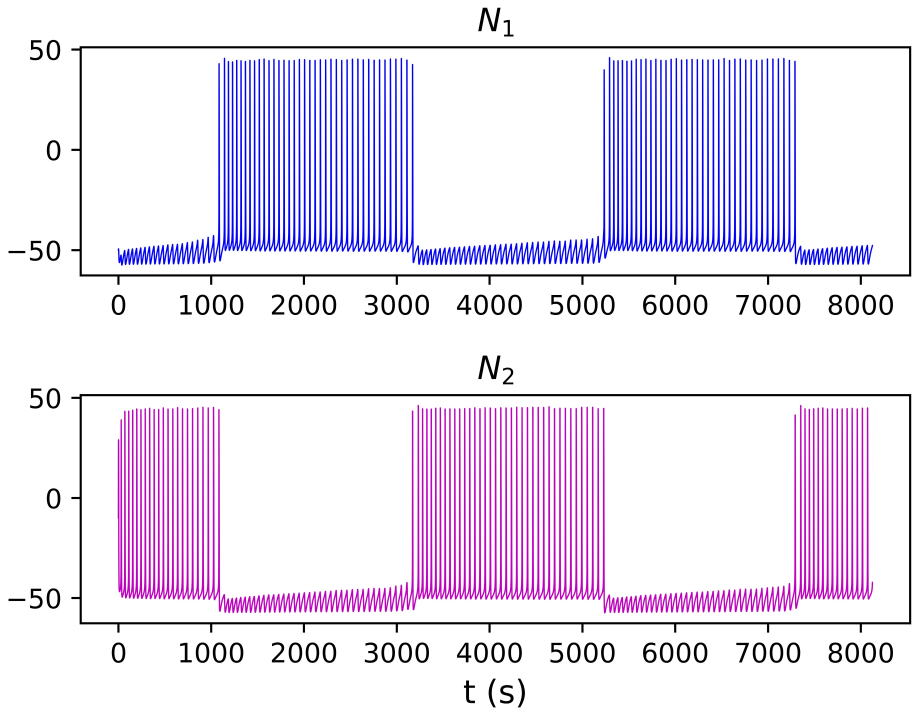

FIGURE 4.5 – Reproduction d'un CPG à partir des neurones de Hodgkin-Huxley via la plateforme biomimétique sur FPGA. Les tensions de membrane ont été multipliées afin de satisfaire les contraintes de précision du CNA.

4.2.5 Simulation de maladies

Plusieurs maladies ainsi que certains symptômes au niveau cellulaire ont été cités dans le Chapitre 1. L'un d'eux concerne la taille des neurones. Ce paramètre fait partie des variables qui peuvent être modifiées en temps réel sur le FPGA. Ainsi, une variation du paramètre de la surface du neurone montre qu'effectivement, elle influe sur la fréquence des potentiels d'action. La figure 1 montre une simulation réalisée à partir du modèle à plusieurs compartiments de [DAVI 2000] et montre que l'augmentation de la taille d'un neurone induit un ralentissement des activités électriques de ce dernier. La figure 4.6 montre une simulation réalisée en utilisant un modèle du bulbe olfactif à plusieurs compartiments [DAVI 2000]. La taille du neurone par défaut est de 16,234 µm. Une diminution de la taille de 20% induit une accélération des activités électriques et une augmentation de 20% implique une

fréquence plus élevée des potentiels d'action.

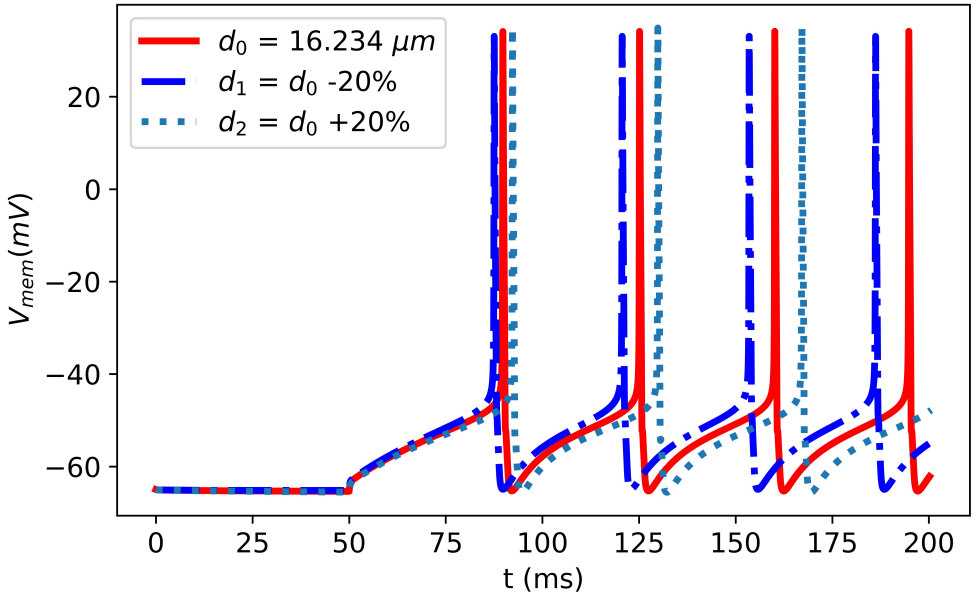

FIGURE 4.6 – Variation de la fréquence de *spike* en fonction de la taille d'un neurone simulé à partir d'un modèle à plusieurs compartiments [DAVI 2000].

Cette simulation ne constitue évidemment pas la reproduction entière et complexe d'une maladie. En revanche, il s'agit là d'un exemple des possibilités qu'offre l'implémentation d'un modèle comme celui de Hodgkin-Huxley. En effet, la simulation vient corroborer les expériences biologiques et peut constituer, lorsqu'elle est utilisée sur un réseau, un puissant outil d'étude pour la communauté.

4.2.6 Interface graphique

Un autre avantage de ce type de plateformes est qu'elles sont flexibles. En effet, il est possible de modifier les paramètres des neurones, comme le montre la figure 4.7, à travers une Interface graphique développée sous Python en utilisant la bibliothèque graphique *tkinter*. L'interface permet non seulement de modifier les paramètres biophysiques des équations de Hodgkin-Huxley (taille du neurone, conductance et potentiel d'équilibre des ions) mais aussi les paramètres de bruit des équations de Ornstein-Uhlenbeck, les courants ioniques et les sorties du CNA.

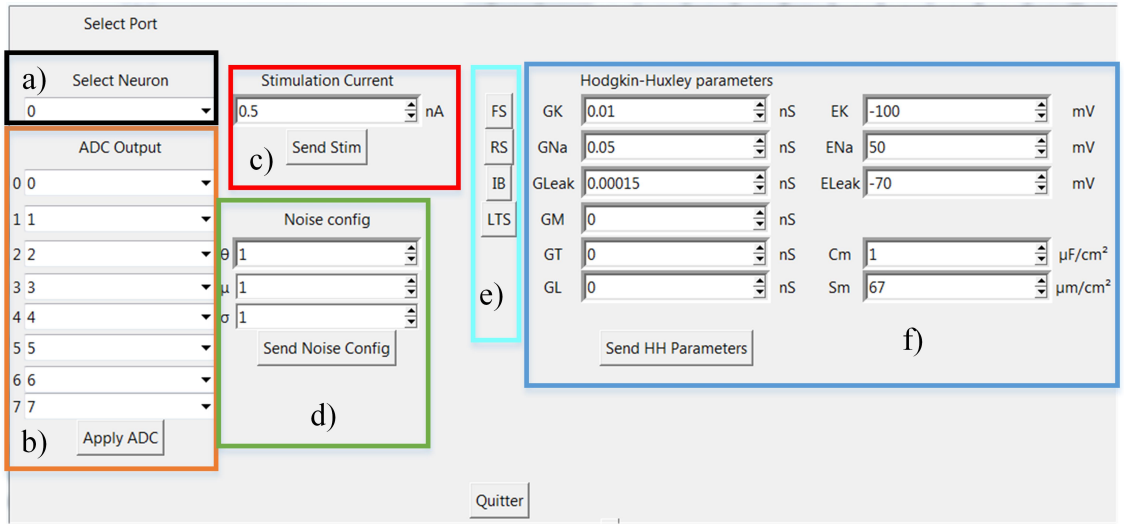

FIGURE 4.7 – Interface graphique développée en utilisant le langage Python pour communiquer avec les plateformes biomimétiques : a) Sélection du neurone sur lequel appliquer les changements; b) Sélection du neurone afin de visualiser sa forme analogique sur une des sorties du CNA; c) Valeur du courant de stimulation en nA à injecter sur le neurone sélectionné en a); d) Paramètres du bruit à appliquer sur les équations de Ornstein-Uhlenbeck; e) Possibilité de rétablir les paramètres FS, RS, IB ou LTS par défaut; f) Paramètres biophysiques des neurones, incluant les conductances des canaux ioniques, les potentiels d'équilibre des ions et la taille des neurones.

4.3 Expériences Biohybrides

Pour rappel, le chapitre 1 a évoqué plusieurs possibilités pour étudier les maladies neurodégénératives. D'un côté, les maladies peuvent être étudiées par simulation des réseaux de neurones dans lesquels on y applique les phénomènes responsables des maladies [LEVI 2017]. D'un autre côté, les expériences biohybrides peuvent permettre de mieux caractériser et modéliser ces phénomènes. Dans ce sens, l'intérêt d'une plateforme de neurones biomimétiques, dont les paramètres sont modifiables en temps réel, est de pouvoir conduire les cellules biologiques dans des états désirés ou afin de reproduire des comportements spécifiques en les interfaçant avec un système de neurones artificiels. Les expériences biohybrides sont présentées dans cette partie en trois phases.

4.3.1 Les organoïdes, une solution pour étudier le cerveau ?

Le cerveau est séparé en différentes régions interconnectées par des axones qui s'étendent d'une région à une autre permettant à l'organe de fonctionner de manière coordonnée. Les axones connectant des régions distantes sont structurellement organisés par des enroulements de faisceaux appelés voies cérébrales [SWAN 2017] [ZING 2014]. Certains de ces faisceaux sont spécifiques à certaines fonctions cérébrales et connectent certaines régions dans un hémisphère comme les régions associées au langage [BERN 2009] [PUJO 2000]. D'autres faisceaux connectent deux hémisphères cérébraux comme le *corpus callosum* qui est le plus large faisceau d'axones du cerveau humain. [YEH 2018] souligne que l'augmentation de nos connaissances sur les structures du cerveau et de son organisation passe par l'étude des faisceaux connectant les régions du cerveau et les réseaux de neurones qui y sont associés.

Organoïdes

Les organoïdes sont des amas de cellules reproduisant l'anatomie d'un organe. Dans notre cas, il s'agit de régions du cerveau. Les organoïdes sont créés par des cellules-souches pluripotentes induites et humaines. Cette technique a été créée par [YAMA 2006], elle lui a valu le prix Nobel en 2012. [KAWA 2017] [KIRI 2019] ont développé des microdispositifs permettant de cultiver les amas cellulaires et de développer des faisceaux d'axones connectant deux organoïdes afin de reproduire le modèle biologique de deux régions connectées du cerveau. Les axones grandissent d'un organoïde à un autre et ce, des deux côtés avant de se rejoindre et finalement former un faisceau.

Imagerie calcium

Depuis la création du *voltage clamp* en 1947, technique développée par Kenneth Cole et George Marmont, de nombreuses méthodes ont été créées pour observer les activités des neurones [BROC 2017]. Dans le cas des organoïdes, il s'agit d'un amas de cellules nerveuses ; il est, de ce fait, difficile d'observer les potentiels d'action et de les attribuer à un neurone en particulier. De plus, le but n'est pas, dans ce cas, d'observer les activités électriques des cellules mais d'observer la connectivité des deux organoïdes par les faisceaux d'axones. L'électroencéphalogramme (EEG) est tout aussi exclu car il ne renseigne pas sur les informations recherchées. Cependant, il existe une autre méthode permettant d'observer les activités des ions calcium dans une cellule et de visualiser ses activités.

L'imagerie calcium est une technique permettant, grâce à un microscope dédié, de déceler les variations du taux de calcium dans une ou plusieurs cellules. Les ions calcium peuvent être observés grâce à un composé fluorescent dont les molécules ont été modifiées et qui émettent un rayonnement à une certaine longueur d'onde lorsqu'il reçoit un rayonnement d'une autre longueur d'onde particulière. Les longueurs d'onde dépendent des molécules choisies qui sont présentées et décrites par [GRIEN 2012]. Ce colorant fluorescent est d'abord préparé et mélangé à une solution physiologique avant d'être mis en contact avec les cultures de cellules biologiques puis placé dans un incubateur. Une fois la période d'incubation finie, le colorant est retiré et les neurones sont placés sous un microscope prévu pour exciter les molécules du colorant avec des lasers possédant les longueurs d'onde nécessaires. Á l'aide d'une caméra spécialisée et d'un logiciel de traitement prévu pour l'imagerie, il est possible d'observer aussi bien sur la caméra que sur le logiciel de traitement, une augmentation de la fluorescence des cellules lorsqu'elles produisent des activités.

Stimulation des organoïdes

Dans notre cas, le composé fluorescent utilisé est le Fluo-4 AM et la caméra, une sCMOS de la marque Andor. Les organoïdes ont été stimulés à partir d'un stimulateur de la marque *multichannel systems* (MCS). Les expériences sur organoïdes ont été menées afin de valider le modèle biologique proposé par [KAWA 2017]. Pour cela, plusieurs étapes sont à valider. Tout d'abord, l'organoïde doit réagir à une stimulation et donc faire apparaître à l'image une variation du taux de calcium en intracellulaire lors d'une stimulation. Ensuite, la gaine d'axones devra pouvoir conduire les influx nerveux depuis l'organoïde stimulé vers l'organoïde non stimulé. Enfin, lorsque les étapes précédentes pourront être validées, il est possible de connecter ces organoïdes à un réseau de neurones artificiel.

Validation de la réponse d'un organoïde Cette étape sera utilisée systématiquement à chaque expérience afin de valider le fonctionnement des organoïdes. Pour cela, le montage de la figure 4.8 a été réalisé pour toutes les expériences qui seront présentées dans la partie consacrée aux organoïdes. Elle représente deux organoïdes connectés (O1 et O2) par des faisceaux d'axones. Les organoïdes sont déposés dans une boîte de Petri sous un microscope équipé de la caméra sCMOS. Cette dernière a été modifiée afin de la connecter au stimulateur qui attendra un signal de la caméra pour lancer des impulsions électriques à intervalles réguliers et pouvoir étudier les corrélations entre la tension de stimulation avec la réponse en calcium.

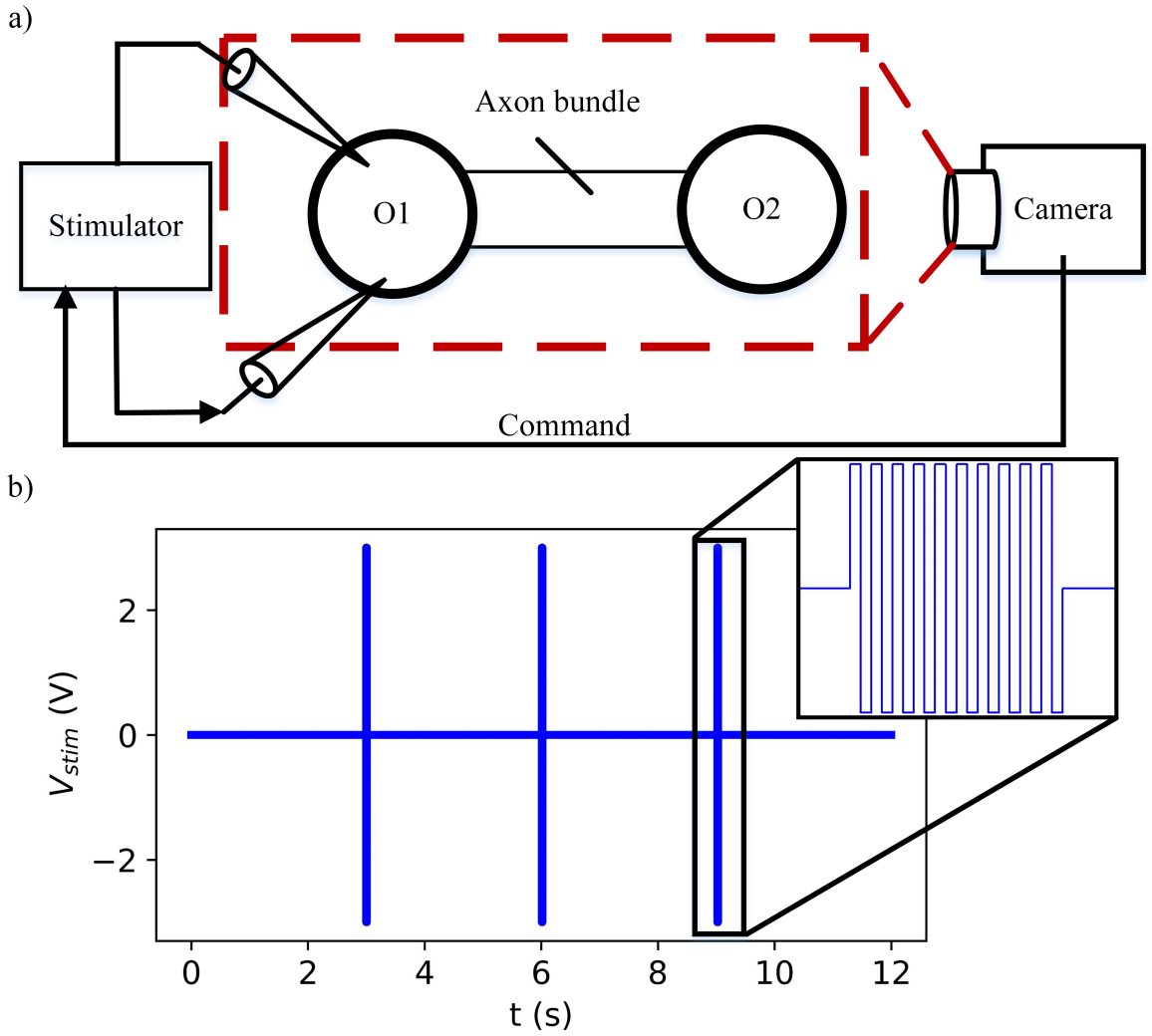

FIGURE 4.8 – a) O1 et O2 sont deux organoïdes connectés par des faisceaux d'axones. Le stimulateur connecté à un organoïde via une électrode, attend de recevoir une information de la caméra afin de démarrer l'enregistrement vidéo et l'envoi d'impulsions ; b) figure représentant un signal biphasique généré par le stimulateur, l'enregistrement de la caméra dure 12 secondes et les impulsions sont séparées par 3 secondes d'intervalle.

Il a été choisi d'utiliser des créneaux de tension biphasiques limités à +/- 3V avec un courant dépendant de la résistance de l'électrode utilisée [MCS 2019], car les expériences ont montré que, d'une part, l'organoïde ne supportait pas une tension supérieure et, d'autre part, qu'il a besoin d'une phase négative afin d'éviter d'endommager l'amas de cellules.

Communication entre le vivant et l'artificiel Les expériences suivantes ont servi à étudier la faisabilité d'une connexion entre le vivant et l'artificiel par l'utilisation de plateformes biomimétiques. Les expériences de la figure 4.9 montrent qu'il est possible de conditionner les cellules par un réseau de neurones artificiel. La figure 4.9.a) représente une plateforme du modèle CMOD A7 à base de FPGA. Il contient un réseau de neurones biomimétiques implémenté par [AMBR 2013] en utilisant le modèle d'Izhikevich. La figure 4.9.c) montre les images d'un organoïde dans deux états différents. Lorsque les cellules sont stimulées, la luminosité augmente suffisamment pour être visible à travers la vidéo renvoyée par la caméra. Dans le cas contraire, l'image reste sombre. Cette variation de luminosité peut être observée sur la figure 4.9.d) montrant une forte corrélation entre les stimulations générées par le CMOD A7 et les variations de luminosité enregistrées par la caméra et le logiciel de traitement. Cette expérience montre qu'il est possible d'amener les cellules à un état dépendant des activités de réseaux de la plateforme biomimétique.

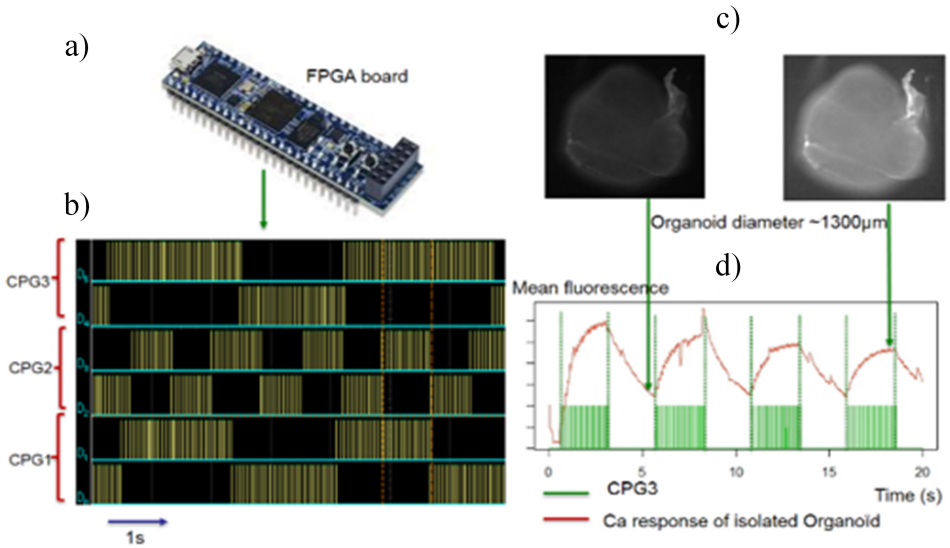

FIGURE 4.9 – Schéma issu de [KHOYb 2019] montrant l'application d'un CPG sur un organoïde et la réponse de ce dernier visualisée par imagerie calcium : a) Plateforme CMOD A7 contenant un FPGA de type Artix-7 et dont les dimensions sont d'environ 18 x 7 cm ; b) Le modèle d'Izhikevich implémenté par [AMBR 2013] génère des impulsions représentant les activités des neurones sur le plan temporel qui peuvent être observées par l'oscilloscope ; c) L'organoïde stimulé est observé par imagerie calcium. Une variation de luminosité est clairement visible lorsque l'organoïde passe de l'état non stimulé à l'état stimulé ; d) Il est possible de corréler la stimulation électrique avec les variations de luminosité montrant qu'il est possible de conduire la cellule à un état conditionné par la plateforme biomimétique.

Un certain nombre de questions amènent la manipulation à être reconsidéré dans le cadre de l'étude d'autres maladies. En effet, les organoïdes sont des amas de cellules, ce qui empêche l'étude de neurones isolés et la compréhension de l'impact de la stimulation sur la structure neurale. De plus, si l'on veut orienter la technologie vers des neuroprothèses, des problématiques de consommation et de dissipation thermiques apparaîtront. Ce questionnement a amené à la mise en place d'un ensemble de systèmes expérimentaux permettant l'étude de cellules isolées.

4.3.2 La microfluidique et la biocompatibilité des matériaux

Les réseaux de neurones sur silicium sont efficaces, car ils permettent de reproduire les activités électriques des neurones et de stimuler ces dernières. Cependant, le manque de compatibilité entre les tissus biologiques et les matériaux conducteurs [BAYL 1999] amène à reconsidérer le système. Une autre branche des neurosciences s'active à réaliser des connexions entre le vivant et l'artificiel par la microfluidique.

[NISHI 2018] a réalisé des systèmes microfluidiques dont la communication avec les cultures de neurones s'effectue par un contrôle rapide des concentrations ioniques par l'utilisation de valves pneumatiques. Les cultures de neurones sont sensibles à l'injection de KCl et montrent une réponse lors des observations sur imagerie calcium. Le matériau utilisé est le Polydiméthylsiloxane (PDMS) afin d'assurer une meilleure biocompatibilité. Des valves contrôlent la pression du flux de KCl afin de contrôler les concentrations ioniques de la solution dans laquelle baignent les neurones.

Un tel système permet de générer des activités électriques semblables aux potentiels d'action générés par un réseau de neurones biologiques en s'affranchissant des problèmes de biocompatibilité. Le réseau de neurones artificiel génère ainsi des activités biomimétiques à une interface microfluidique afin de communiquer avec les neurones biologiques afin d'établir une communication avec ceux-ci.

La figure 4.10.a) montre l'état des neurones, par leur observation à travers l'imagerie calcium, lorsque les cellules nerveuses ne sont pas stimulées et lorsqu'elles le sont. Le schéma global de la manipulation peut être observé à travers la figure 4.10.b et montre un système électronique biomimétique (le FPGA) connecté au système microfluidique contenant les cultures de neurones. L'intérêt d'un tel système est de permettre une connexion entre un réseau de neurones complexe et bioréaliste à des cellules biologiques en faisant abstraction de la problématique liée à la biocompatibilité.

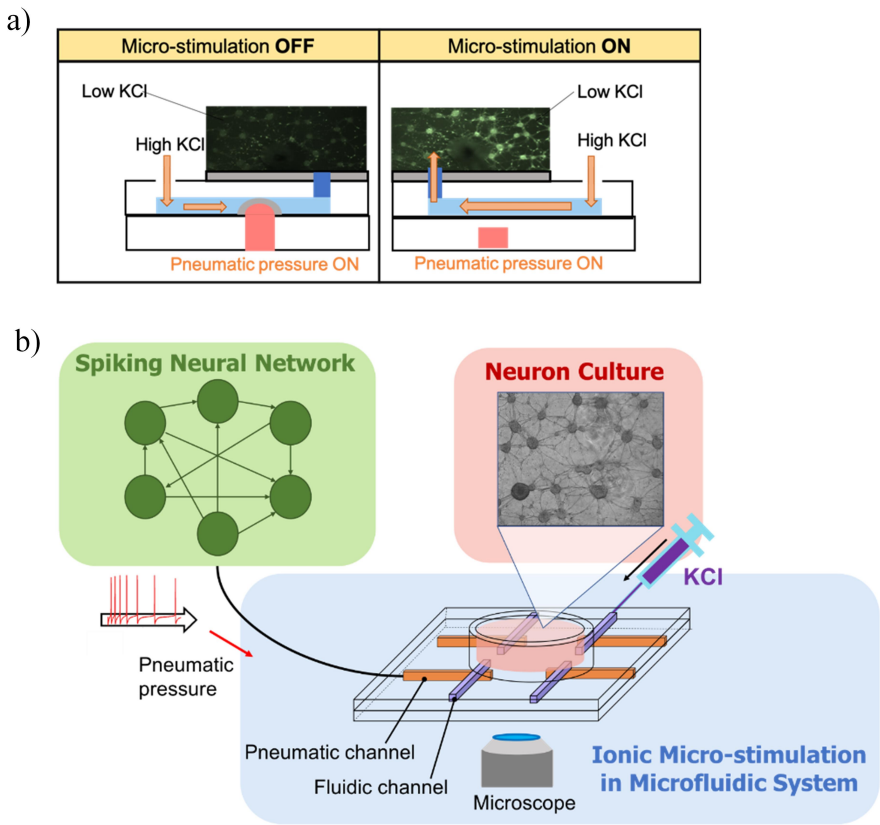

FIGURE 4.10 – Couplage du réseau de neurones artificiel avec le système microfluidique.

4.3.3 Système expérimental pour l'étude de cellules biologiques

Un système expérimental a été développé basé sur un besoin de flexibilité du réseau de neurones biomimétiques et de l'observation de cellules isolées. La figure 4.11 représente une plateforme électronique à base de FPGA reproduisant artificiellement des neurones à simple compartiment ou à plusieurs compartiments. La plateforme est connectée via un DAC à une carte d'adaptation de tension. Cette dernière est utile afin de réduire la tension de sortie du DAC pour le mettre à l'échelle des potentiels d'action. De plus, la carte permet aussi d'augmenter la tension dans le cadre de l'étude d'autres cellules telles que les cardiomyocites qui ne seront pas détaillées ici. La plateforme d'adaptation est ensuite connectée à un ou plusieurs MEA contenant les cellules à étudier.

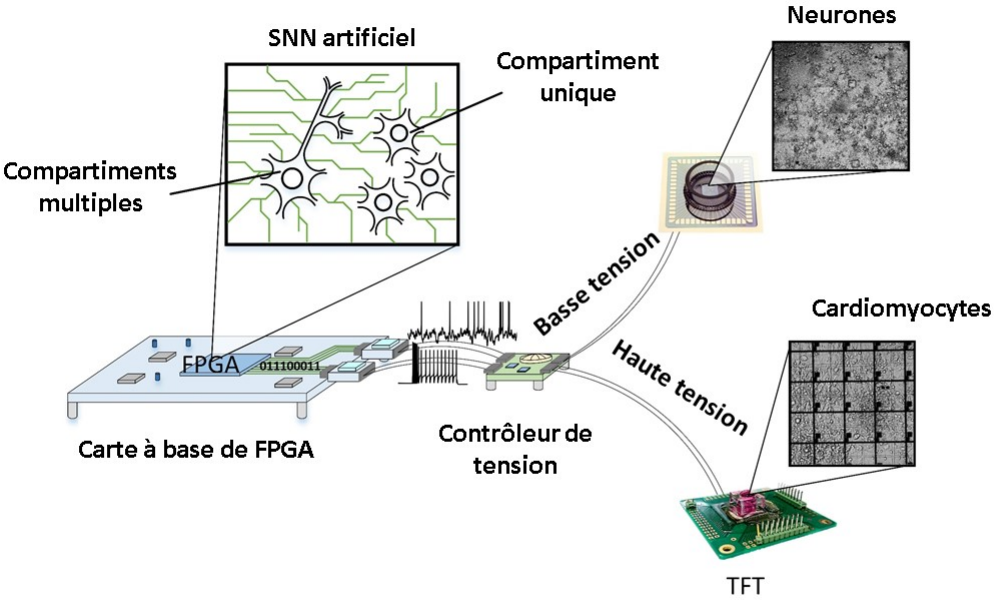

FIGURE 4.11 – Représentation du système expérimental permettant de faire interagir un système à base de *Spiking Neural Network* (SNN) contenant des neurones à plusieurs compartiments ou des neurones à compartiment unique avec des cellules biologiques cultivées sur MEA. Les neurones artificiels sont implémentés sur FPGA. Les données numériques du FPGA sont converties via un DAC en signaux analogiques. L'amplitude de ces signaux est ensuite adaptée afin de reproduire les mêmes tensions que les cellules biologiques.

Ce type de système nécessite plus d'expériences afin de valider son utilisation avant de conclure sur l'étude du comportement des cellules et la communication entre le vivant et l'artificiel. Il permettra cependant de déterminer l'importance des paramètres dans un réseau de neurones. En effet, le modèle à plusieurs compartiments permet, en plus d'inclure les paramètres de morphologie et de Hodgkin-Huxley, de reproduire une forme plus précise de potentiels d'action. De ce fait, ce système est capable d'approfondir l'étude portant sur l'importance et les erreurs admissibles, lors de la communication intercellules, sur les dimensions spatiale, temporelle et sur la forme du potentiel d'action.

4.4 Application de la bibliothèque VHDL pour ingénierie neuromorphique

Les équations différentielles des modèles de neurones sont très similaires à d'autres modèles mathématiques. La bibliothèque VHDL conçue et montrée dans le chapitre 3 peut être adaptée à d'autres applications dont les algorithmes et modèles sont proches du neuromorphique. Cette section a pour but de montrer que la bibliothèque peut non seulement s'ouvrir à d'autres domaines mais qu'elle permet en plus de fournir des résultats intéressants. Dans un premier temps, cette partie décrira l'utilisation de ces méthodes dans l'implémentation de réseaux biochimiques. Dans un second temps, une partie sera consacrée à la conception d'un système intégrant un algorithme basé sur les modèles de neurones qui permet de

résoudre des problèmes d'optimisation combinatoire.

4.4.1 Modèle proie-prédateur

Le modèle proie-prédateur ainsi que les équations qui y sont associés sont présentés dans l'annexe 3. Les équations ont été implémentées sur FPGA [TANH 2019] en utilisant les méthodes proposées dans le chapitre 3 et permettent d'obtenir les résultats de la figure 4.18. L'implémentation a été effectuée avec des valeurs encodées sur 4 bits de partie entière et 32 bits de partie décimale offrant une grande précision, car ces systèmes ne sont pas très stables. Les multiplications, les divisions et les équations différentielles utilisent respectivement les DSP48E1, l'algorithme non-restaurant et la méthode d'Euler avec une précision de 2^{-6}. Les figures 4.18.a) et 4.18.b) représentent les résultats de la simulation FPGA sur oscilloscope des oscillations typiques du système proie-prédateur en utilisant différents paramètres.

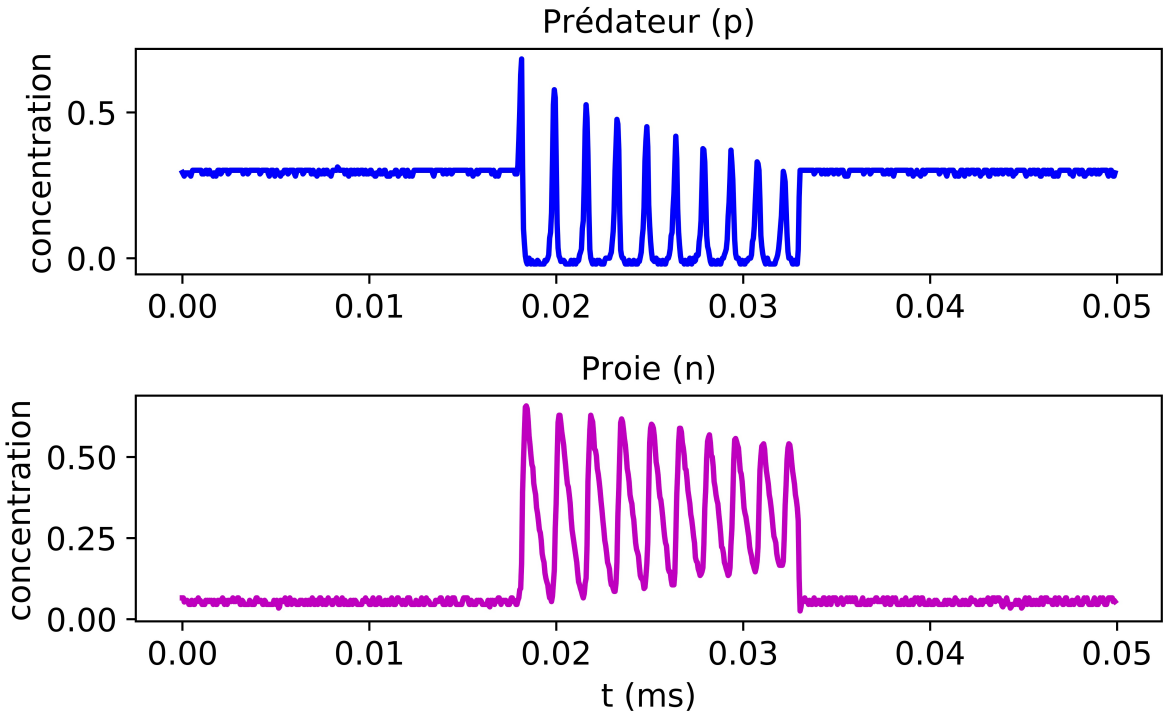

FIGURE 4.12 – Résultat mesuré via un oscilloscope lors de la simulation sur FPGA de la dynamique du système proie-prédateur. Les courbes montrent une dynamique similaire à celle simulée sur ordinateur.

La réalisation d'un tel système permet, comme pour le cas des réseaux de neurones, d'assister la recherche biologique en cherchant des architectures de réseaux biochimiques et des paramètres de ces réseaux qui correspondent aux observations. Elle permettrait aussi de contribuer à la recherche de réseaux génériques permettant de remplir une fonction donnée (par exemple la régulation du niveau de glucose dans le sang).

4.4.2 Problème d'optimisation combinatoire

Les problèmes d'optimisation combinatoire sont étudiés dans de nombreux domaines, ils font l'objet d'une plus longue explication dans l'annexe 4.1. Un des algorithmes d'optimisation combinatoire a été implémenté sur FPGA (voir annexe 4.2) et montre les résultats qui ont été comparés aux différentes méthodes et résultats de l'état de l'art sur des dispositifs différents :

— «*break-out local search* » (BLS) [BENLI 2013] qui a été utilisé pour trouver la plupart des solutions de références à partir des paramètres de benchmark de l'université de Stanford [1]. La simulation est réalisée sur un seul cœur de CPU ;

— « *Parrallel tempering* » (ou replica exchange *Monte-Carlo Markov chain sampling*) (PT) [HUKU 1996] ;

— «*Normalized mean field annealing*» (NMFA) [KING 2018] implémenté pour être calculé sur GPU.

La figure 4.13 affiche les résultats de l'implémentation de la méthode proposée par [LELEU 2019] sur FPGA. Ils représentent la vitesse pour résoudre un problème et la consommation du système en fonction de la taille du problème qui correspond à la taille d'une matrice. La figure 4.13.a) montre que malgré une plus grande vitesse de la méthode NMFA pour les petites instances, les solutions proposées offrent de meilleurs résultats lorsque les tailles des problèmes augmentent. De plus, la figure 4.13.b montre que le système proposé possède une consommation inférieure aux systèmes de l'état de l'art. La carte de développement de la marque avenet contenant le FPGA Kintex 7 KU040 Ultrascale possède des régulateurs intégrant des capteurs de tension, de courant et de puissance. Il est possible d'accéder à ces informations en se connectant aux régulateurs et en déchiffrant les trames. Les énergies de l'état de l'art correspondent à la consommation affichée dans la littérature multipliée par les temps de calcul pour chaque taille de problème. Cependant, peu d'information apparaissent sur la méthodologie employée pour mesurer les consommations de l'état de l'art.

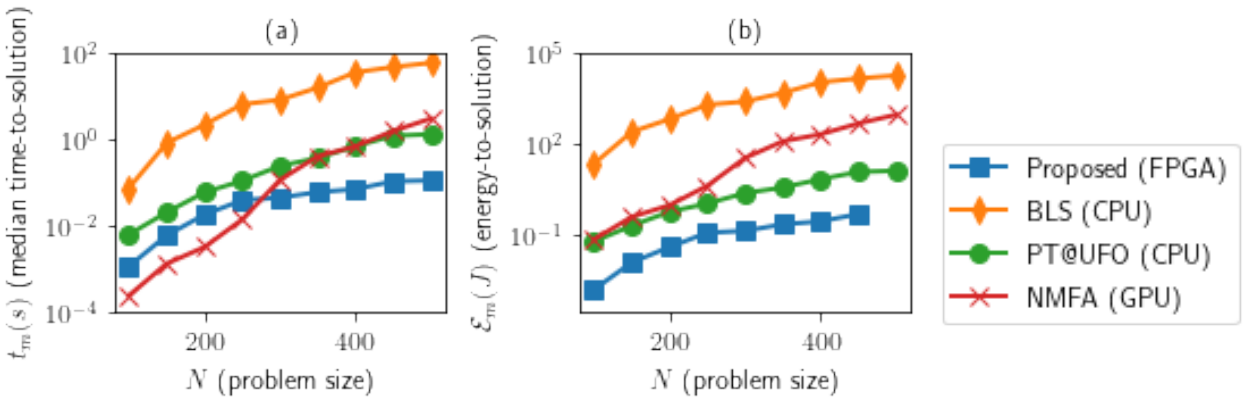

FIGURE 4.13 – Résultat de l'implémentation FPGA en effectuant les mesures sur plusieurs configurations différentes en comparant l'état de l'art avec l'implémentation proposée ; a) Le temps moyen que met le système pour atteindre la solution optimale ; b) La consommation des différents systèmes.

1. https ://web.stanford.edu/~yyye/yyye/Gset/

4.5 Résumé

Les plateformes de réseaux de neurones biomimétiques ont été développées sur la base de bibliothèques VHDL génériques de fonctions mathématiques. Chaque niveau de détail (neurone, synapse, bruit et réseau) a été validé. Ensuite cette plateforme de réseaux de neurones a été utilisée lors d'expériences biohybrides. Les premières expériences relatent le fonctionnement et la validation d'un modèle biologique d'organoïdes connectés par des faisceaux d'axones. Les expériences sur ces amas de cellules ont montré qu'il est possible de communiquer entre un réseau de neurones artificiel et les cellules vivantes. Nous avons ensuite étudié la problématique de biocompatibilité entre système électronique et cellules. De ce fait, une solution portée sur un couplage entre la plateforme à base de FPGA et une plateforme microfluidique afin de stimuler les neurones de manière chimique. Le couplage a pu montrer que le FPGA peut en effet piloter la plateforme microfluidique qui, elle-même, permet de stimuler les cellules. Enfin, les fonctions réalisées dans la bibliothèque VHDL ont été utilisées afin de réaliser des systèmes de calculs utilisant des algorithmes dont la dynamique est similaire à celle produite par les réseaux de neurones. D'une part, les fonctions peuvent être utilisées dans des modèles de «*DNA computation*» et, d'autre part, dans les résolutions de problèmes d'optimisation combinatoire, elles montrent de bons résultats.

Conclusion

La problématique générale de ce manuscrit concernait l'utilisation d'une plateforme configurable temps réel contenant un ou plusieurs réseaux de neurones pour des applications biomédicales. Cette approche a été étudiée sous deux angles. La première possibilité était la simulation d'un réseau de neurones bioréalistes et biodétaillés sur lequel on applique des perturbations similaires aux troubles des maladies neurodégénératives. La deuxième possibilité était l'interfaçage du ou des réseaux de neurones artificiels avec des cellules biologiques. Cette méthode permet d'étudier les comportements des neurones en profitant de la flexibilité que peut offrir un réseau de neurones artificiel sur plateforme numérique et d'offrir des stimulations adaptatives pour des protocoles thérapeutiques bioélectriques. Ces deux méthodes constituent une première étape vers la conception de neuroprothèses. Pour répondre à la problématique, ce manuscrit s'organisait autour de 4 chapitres permettant d'introduire le contexte, de développer les parties ayant trait à la conception d'une plateforme biomimétique et finalement les applications.

Ce manuscrit a tout d'abord introduit les fonctions de certaines cellules du système nerveux et leur rôle ainsi que la manière dont les neurones communiquent entre eux. Cette base a permis d'introduire les maladies neurologiques et les troubles, au niveau cellulaire, qu'elles infligent aux malades. Elle a permis aussi de décrire les réseaux de neurones biomimétiques. Le système nerveux a été modélisé par différentes méthodes qui ont été classées par [IZHI 2003] en fonction des coûts d'implémentation et de leur niveau de détails. Il en ressort que le modèle de Hodgkin-Huxley est le plus adapté pour la simulation de neurones biomimétiques. Cependant, ce modèle possède un certain nombre de limitations représentées par une simplification du fonctionnement des neurones et par le manque de certains phénomènes. De ce fait, un modèle à plusieurs compartiments est plus adapté, ce qui a été retenu afin d'atteindre le but recherché.

L'utilisation d'une plateforme comme le *Field Programmable Gate Array* (FPGA) permet d'optimiser les calculs et d'atteindre des performances supérieures à ce qu'un CPU ou un GPU pourrait afficher. Des calculs multiparallèles et *pipeline* permettent au FPGA, en plus de sa flexibilité, d'être une des plateformes embarquées offrant le plus de possibilités. Cependant, une implémentation sur une telle plateforme requiert de prendre en compte les limitations technologiques des FPGA. Les équations doivent donc être adaptées par diverses méthodes à une implémentation numérique leur permettant d'être moins coûteuse en ressources et rapides à calculer. Les méthodes de régression et d'interpolation sont celles qui ont été présentées et ont montré de bons résultats qui ont été validés par des outils comme le «*Pearson Product Moment Correlation*» (PPMC), la «*Cross-correlation*» et l'analyse de bifurcations.

L'optimisation des équations de Hodgkin-Huxley peut être aussi effectuée en prenant en compte les possibilités offertes par le FPGA ce qui a donné naissance à une bibliothèque VHDL optimisée pour l'ingénierie neuromorphique. Cette bibliothèque régit l'ensemble des techniques d'optimisation et d'implémentation présentées allant de l'encodage des données jusqu'aux opérateurs mathématiques permettant d'effectuer des calculs rapides et à bas coût. Nous avons listé la possibilité d'effectuer une division en un cycle d'horloge avec la méthode non-restaurante, la multiplication par le biais des Digital Signal Processing (DSP) qui sont des composants multifonctions présents dans beaucoup de FPGA, la méthode d'Euler pour la résolution d'équations différentielles, les méthodes de calcul de produits de matrice par vecteur pour le calcul des synapses par exemple et le *COordinate Rotation DIgital*

Computer (CORDIC) pour les fonctions hyperboliques et exponentielles. Ce dernier possède une alternative possédant un plus grand potentiel pour le calcul des tangentes hyperboliques qui est l'interpolation par les polynômes de Lagrange.

L'utilisation de ces méthodes a permis la création d'une plateforme biomimétique dont les performances surpassent l'état de l'art en reproduisant un réseau de neurones entièrement connecté par des synapses et dont les paramètres peuvent être modifiés en temps réel. De plus, différents types de neurones ont été implémentés pour générer des comportements très similaires à la biologie. Une option permettant d'ajouter du bruit synaptique a été implémentée permettant de générer, via le FPGA, des activités spontanées similaires à celles mesurées dans le cortex d'un chat ou le néocortex d'un rat. Ce biomimétisme des neurones artificiels a permis, entre autres, de reproduire des activités de réseaux comme le *Central Pattern Generator* (CPG) qui peut être retrouvé dans les réseaux de la moelle épinière. Les expériences biohybrides visant à établir une connexion entre la plateforme biomimétique et les cellules vivantes ont montré qu'il est possible de piloter les cellules biologiques par un réseau artificiel. En effet, l'imagerie calcium sur des cellules excitées par un réseau de neurones générant un CPG a montré de fortes corrélations entre la réponse calcique des cellules et la stimulation électrique représentant un CPG en sortie du *Spiking Neural Network* (SNN). Le problème d'une connexion directe entre un système électrique et un système biologique est la biocompatibilité. De ce fait, d'autres expériences ont montré qu'il était possible et efficace de piloter une plateforme microfluidique qui excite chimiquement les neurones. Cette méthode permet de s'affranchir des problématiques citées.

Dans une perspective plus large, ces travaux de thèse ont permis, grâce à l'élaboration de la bibliothèque VHDL contenant un certain nombre de méthodes, d'ouvrir l'utilisation des FPGA à une autre application que les réseaux de neurones. Des résultats ont montré l'efficacité de la plateforme numérique à simuler des réseaux biochimiques mettant en relation les aspects génétiques entre un prédateur et sa proie. La bibliothèque a aussi permis de développer un outil dépassant, là-aussi, l'état de l'art pour la résolution de problèmes liés à l'optimisation combinatoire.Ces travaux de thèse sont les prémices de futurs projets concernant l'étude du comportement du système nerveux.

Des points d'amélioration de la plateforme sont possibles. Tout d'abord, sans avoir la prétention de recréer un système nerveux artificiel complet dans tous ses détails, certaines parties de cellules et certains phénomènes importants à l'étude des maladies neurodégénératives pourraient être implémentés. Il conviendrait d'augmenter le détail des neurones en ajoutant les cellules gliales et la myéline qui sont la cible de détériorations provoquées par certaines maladies. La plasticité à long terme comme la «*Spike Timing Dependant Plasticity*» (STDP) est un ajout qui permettrait une meilleure correspondance avec les dynamiques biologiques.

Enfin, pour établir une connexion plus efficace entre un système de neurones artificiels et un réseau de neurones biologiques, il faudrait une interface bidirectionnelle temps réel. Cette interface est en cours d'élaboration avec un système électronique permettant une détection et un *spike sorting* en temps réel. Les *Micro Electrodes Array* (MEA) sont les outils le plus communément utilisés, cependant la qualité du signal enregistré et la taille de l'électrode ne permettent pas une approche « *single cell* » et une modélisation détaillée. D'autres outils comme les *Thin Film Transistor* (TFT) développés par la Dr Agnès Tixier-Mita [TIXIER

2019] de l'Université de Tokyo et des matrices de nanoprobes développées par le Dr Guilhem Larrieu au LAAS [CASAN 2018] pourraient correspondre à ce besoin. De plus, il est possible d'effectuer des simulations numériques pour l'interface biophysique afin d'assister le développement de ces types de plateformes [BOURJ 2017].

Bibliographie

[ABEL 1993] M. Abeles, H. Bergman, E. Margalit and E. Vaadia, 1993, Spatiotemporal firing patterns in the frontal cortex of behaving monkeys, Journal of Neurophysiology, 70 :1629-1638, Doi : 10.1152/jn.1993.70.4.1629

[ABEL 2001] M. Abeles and I. Gat, 2001, Detecting precise firing sequences in experimental data, Journal of Neuroscience Methods, 107 :141-154, Doi : 10.1016/S0165-0270(01)00364-8

[ADRI 1926] E. D. Adrian and Y. Zotterman, 1926, The impulses produced by sensory nerve endings, The Journal of physiology, 61 :151-171, Doi : 10.1113/jphysiol.1926.sp002281

[AHME 2017] A. M. Abdelsalam, J. M. P. Langlois and F. Cheriand, 2017, A Configurable FPGA Implementation of the Tanh Function Using DCT Interpolation, IEEE, 25th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), Doi : 10.1109/FCCM.2017.12

[AKBA 2018] K. Akbarzadeh-Sherbaf, B. Abdoli, S. Safari and A.H. Vahabie, 2018, A Scalable FPGA Architecture for Randomly Connected Networkd of Hodgkin-Huxley Neurons, Frontiers in Neuroscience, 12 :698, Doi : 10.3389/fnins.2018.00698

[ALBA 2018] S. J. Van Albada, A. G. Rowley, J. Senk, M. Hopkins, M. Schmidt, A. B. Stokes, D. R. Lester, M. Diesmann and S. B. Furber, 2018, Performance Comparison of the Digital Neuromorphic Hardware SpiNNaker and the Neural Network Simulation Software NEST for a Full-Scake Cortical Microcircuit Model, Frontiers in Neuroscience, 12 :291, Doi : 10.3389/fnins.2018.00291

[AMBR 2013] M. Ambroise, T. Levi, S. Joucla, B. Yvert and S. Saïghi, 2013, Real-time biomimetic Central Pattern Generators in an FPGA for hybrid experiments, Frontiers in Neuroscience, 7 :215, Doi : 10.3389/fnins.2013.00215

[AMBR 2015] M. Ambroise, 2015, Hybridation des réseaux de neurones : de la conception du réseau à l'interopérabilité des systèmes neuromorphiques, Thèse

[ANDE 1992] B. B. Andersen, L. Korbo and B. Pakkenberg, 1992, A quantitative study of the Human Cerebellum with Unbiased Stereological Techniques, Journal of Comparative Neurology, 326 :549-560, Doi : 10.1002/cne.903260405

[ARTH 2012] J. V. Arthur, P. A. Merolla, F. Akopyan, R. Alvarez, A. Cassidy, S. Chandra, S. K. Esser, N. Imam, W. Risk, D. B. D. Rubin, R. Manohar and D. S. Modha, 2012, Building block of a programmable neuromorphic substrate : A digital neurosynaptic core, IEEE, The 2012 International Joint Conference on Neural Networks (IJCNN), Doi : 10.1109/IJCNN.2012.6252637

[BATE 2013] H. S. Bateup, C. A. Johnson, C. L. Denefrio, J. L. Salunier, K. Komacker and B. L. Sabatini, 2013, Excitatory-Inhibitory Synaptic Imbalance Leads to Hippocampal Hyperexcitability in Mouse Models of Tuberous Sclerosis, Neuron Cell press , 78 :510-522, Doi : 10.1016/j.neuron.2013.03.017

[BAYL 1999] S. C. Bayliss, L. D. Buckberry, I. Flandcher and M. J. Tobin, 1999, The culture of neurons on silicon, Sensors and Actuatoors A : Phsyical, 74 :139-142, Doi : 10.1016/S0924-

4247(98)00346-X

[BENLI 2013] U. Benlic and J-K. Hao, 2013, Breakout local search for the quadratic assignment problem, Applied Mathematics and Computation, 219 :4800-4815, Doi : 10.1016/j.amc.2012.10.106

[BERN 2009] B. Bernal and A. Ardila, 2009, The role of the arcuate fasciculus in conduction aphasia, Brain, 132 :2309-2316, Doi : 10.1093/brain/awp206

[BILB 1985] G. Bilbro, R. Mann, T. K. Miller, W. E. Snyder, D. E. Van Den Bout and M. White, 1989, Optimization by mean field annealing, Advances in Neural Information Processing Systems, 91-98

[BINC 2006] S. Binczak, S. Jacquir, J. M. Bilbault, V. Kasantsev and V. Nekorkin, 2006, Experimental study of electrical FitzHugh-Nagumo neurons with modified excitability, Neural Networks, 19 :648-693, Doi : 10.1016/j.neunet.2005.07.011

[BLAN 2019] D. Blanchard, K. Aihara and T. Levi, 2019, Snake Robot Controlled by Biomimetic CPGs, Journal of Roboticsn Networking and Artificial Life, 5 :253-256, Doi : 10.2991/jrnal.k.190220.010

[BLISS 1973] T. V. Bliss and T. Lomo, 1973, Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path, The Journal of physiology, 232 :331-356, Doi : 10.1113/jphysiol.1973.sp010273

[BLUM 2003] C. Blum and A. Roli, 2003, Metaheuristics in combinatorial optimization : Overview and conceptual comparison, ACM Computing Surveys, 35 :268-308, Doi : 10.1145/937503.937505

[BMEL 2006] B. W. Mel, 2006, Why Have Dendrites - A Computational Perspective, Oxford University Press, 16, Doi : 10.1093/acprof :oso/9780198566564.001.0001

[BONA 2014] S. Y. Bonabi, H. Asgharian, S. Safari and M. N. Ahmadabadi, 2014, FPGA implementation of a biological neural network based on the Hodgkin-huxley neuron model, Frontiers in Neuroscience , 8 :379, Doi : 10.3389/fnins.2014.00379

[BONI 2013] P. Bonifazi, F. Difato, P. Massobrio, G. L. Breschi, V. Pasquale, T. Levi, M. Goldin, Y. Bornat, M. Tedesco, M. Bisio, S. Kanner, R. Galron, J. Tessadori, S. Taverna and M. Chiappalone, 2013, In vitro large-scale experimental and theoretical studies for the realization of bidirectional brain-prostheses, Frontiers in Neural Circuit , 7 :40, Doi : 10.3389/fncir.2013.00040

[BONO 2017] J. Bono and C. Clopath, 2017, Modeling somatic and dendritic spike mediated plasticity at the single neuron and network level, Nature Communication, 8 :706, Doi : 10.1038/s41467-017-00740-z

[BOURJ 2017] A. Bourjilat, F. Sarry, D. Kourtiche and M. Nadi, 2017, Modelization of interdigitated electrode sensor for impedance spectroscopy measurement, IEEE, Eleventh Inter-

national Conference on Sensing Technology (ICST), Doi : 10.1109/ICSensT.2017.8304461

[BOWE 1994] J. M. Bower, 1994, Control of sensory data acquisition, International review of neurobiology, 41 :489-513, Doi : 10.1016/S0074-7742(08)60367-0

[BRET 2005] R. Brette and W. Gerstner, 2005, Adaptive Exponential Integrate-and-Fire Model as an Effective Description of Neuronal Activity, Journal of Neurophysiology, 94 :3637-3642, Doi : 10.1152/jn.00686.2005

[BRETa 2015] R. Brandte, 2015, What is the most realistic single compartment model of spike initiation ?, PLOS, 11 :4, Doi : 10.1371/journal.pcbi.1004114

[BRETb 2015] R. Brette , 2015, Philosophy of the Spike : Rate-Based vs. Spike-Based Theories of the Brain, Frontiers in Systems Neuroscience, 9 :151, Doi : 10.3389/fnsys.2015.00151

[BRIT 1996] K.H. Britten, W. T. Newsome, M. N. Shadlen, S. Celebrini and J.A. Movshon, 1996, A relationship between behavioral choice and the visual responses of neurons in macaque MT, Cambridge University Press, Visual Neuroscience, 13 :87-100, Doi : 10.1017/S095252380000715X

[BROC 2017] F. D. Broccard, S. Joshi, J. Wang and G. Cauwenberghs, 2017, Neuromorphic neural interfaces : from neurophysiological inspiration to biohybrid coupling with nervous systems, Journal of Neural Engineering, 14 :42, Doi : 10.1088% 2F1741-2552% 2Faa67a9}

[BUCCE 2019] S. Buccelli, Y. Bornat, I. Colombi, M. Ambroise, L. Martines, V. Pasquale, M. Bisio, J. Tessadori, P. Nowak, F. Grassia, A. Averna, M. Tedesco, P. Bonifazi, F. Difato, P. Massobrio, T. Levi and M. Chiappalone, 2019, A neuroprosthetic sytem to restore neuronal communication in modular networks, iScience, 19 :402-414, Doi : 10.1101/514836

[CARM 2003] J. M. Carmena, M. A. Lebedev, R. E. Crist, J. E. OĎoherty, D. M. Santucci, D. F. Dimitrov, P. G. Patil, G. S. Henriquez and M. A. L. Nicolelis, 2003, Learning to Control a Brain–Machine Interface for Reaching and Grasping by Primates, PLOS Biology, 1 :42, Doi : 10.1371/journal.pbio.0000042

[CASAN 2018] A. Casanova, L. Bandtamin, M-C. Blatche, F. Mathieu, H. Martin, D. Gonzalez-Dunia, L. Nicu and G. Larrieu, 2018, Nanowire based bioprobes for electrical monitoring of electrogenic cells, Journal of Physics : Condensed Matter, 30 :46, Doi : https ://doi.org/10.1088/1361-648X/aae5aa

[CASS 2011] A. Cassidy and A. G. Andreaou, 2011, Design of a One Million Neuron Single FPGA Neuromorphic System for Real-time Multimodal Scene Analysis, IEEE, 45th Annual Conference on Information Sciences and Systems, Doi : 10.1109/CISS.2011.5766099

[CHAN 2001] L. Chang and M. Karin, 2001, Mammalian MAP kinase signalling cascades, Nature, 410 :37-40, Doi : 10.1038/35065000

[CHIKL 2002] D. B. Chiklovskii, T. Schikorsko and C. F. Stevens, 2002, Wiring Optimization

in Cortical Circuit, Neuron, Cell Press, 34 :341-347, Doi : 10.1016/S0896-6273(02)00679-7

[CHOU 2014] Z. Chou, J. Lim, S. Brown, M. Keller, J. Bugbee, F. D. Broccard, M. L. Khraiche, G. A. Silva, G. cauwenberghs, 2014, Bidirectional neural interface : Closed-loop feedback control for hybrid neural systems, IEEE, 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Doi : 10.1109/EMBC.2015.7319258

[COHE 2013] M. R. Cohen and A. Kohn, 2011, Measuring and interpreting neuronal correlations, Nature Neuroscience, 14 :811-819, Doi : 10.1038/nn.2842

[CORRE 2009] N. M. Correa, Y. Lin T. Adali and V. D. Calhoun, 2009, Fusion of FMRI, SMRI, and EEG Data Using Canonical Correlation Analysis, IEEE, International Conference on Acoustics, Speech and Signal Processing, Doi : 10.1109/ICASSP.2009.4959601

[COST 2016] V. Costa, S. Aigner, M. Vukcevic, E. Sauter, K. Behr, M. Ebeling, T. Dunkley, A. Friedlein, S. Zoffmann, C. A. Meyer, F. Knoflach, S. Lugert, C. Patsch, F. Fjeldskaar, L. Chicha-Gaudimier, A. Kiialainen, P. Piraino, M. Bedoucha, M. Graf, S. Jessberger, A. Ghoshn J. Bischofberger and R. Jagasia, 2016, mTORC1 Inhibition Corrects Neurodevelopmental and Synaptic Alterations in a Human Stem Cell Model of Tuberous Sclerosis, Cell Press, Cell report, 15 :86-95, Doi : 10.1016/j.celrep.2016.02.090

[DAUE 2003] W. Dauer and S. Przedborski, 2003, Parkinson's Disease : Mechanisms and Models, Cell Press, Neuron, 39 :889-909, Doi : 10.1016/S0896-6273(03)00568-3

[DAVI 2000] A. P. Davison, J. Feng and D. Brown, 2000, A reduced compartmental model of the mitral cell for use in network models of the olfactory bulb, Brain Research Bulletin , 51 :393-399, Doi : 10.1016/S0361-9230(99)00256-7

[DAVI 2018] M. Davies, N. Srinivasa, T-H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, Y. Liao, C-K. Lin, A. Lines, R. Liu, D. Mathaikutty, S. McCoy, A. Paul, J. Tse, G. Venkataramanan, Y-H. Weng, A. Wild, Y. Yang and H. Wang, 2018, Loihi : A Neuromorphic Manycore Processor with On-Chip Learning, IEEE, 38 :82-99, Doi : 10.1109/MM.2018.112130359

[DEBA 2011] D. Debanne, E. Campanac, A. Bialowas, E. Carlier and G. Alcaraz, 2011, Axon Physiology, Physiological Reviews, 91 :555-602, Doi : 10.1152/physrev.00048.2009

[DELO 2005] M. DeLorimier and A. DeHon, 2005, Floating-point sparse matrix-vector multiply for FPGAs, Proceeding, ACM/SIGDA 13th international symposium on Field-programmable gate arrays, Doi : 10.1145/1046192.1046203

[DEST 2001] A. Destexhe, M. Rudolph, J-M. Fellous and T. J. Sejnowski, 2001, Fluctuating synaptic conductances recreate in vivo-like activity in neocortical neurons, Neuroscience, 107 :13-24, Doi : 10.1016/S0306-4522(01)00344-X

[DEST 1998] A. Destexhe, Z. F. Mainen and T. J. Sejnowski, 1998, Kinetic Models of Synaptic Transmission, MIT Press, Cambridge, 46023

[ELOW 2000] M. B. Elowitz, 2000, A synthetic oscillatory network of transcriptional regulators, Nature, 403 :335-338, Doi : 10.1038/35002125

[FABE 1989] D. S. Faber and H. Korn, 1989, Electrical Field Effects : Their Relevance in Central Neural Networks, Physiological Reviews, 69 :821-863, Doi : 10.1152/physrev.1989.69.3.821

[FOX 2006] M. D. Fox, M. Corbandta, A. Z. Snyder, J. L. Vincent and M. E. Raichle, 2006, Spontaneous neuronal activity distinguishes human dorsal and ventral attention systems, PNAS , 103 :26, Doi : 10.1073/pnas.0604187103

[FROE 2005] R. C. Froemke, M. Pee and Y. Dan, 2005, Spike-timing-dependent synaptic plasticity depends on dendritic location, Nature, 434 :221–225, Doi : 10.1038/nature03366

[FUJI 2013] T. Fujii and Y. Rondelez, 2013, Predator–Prey Molecular Ecosystems, ACS Publications, 7 :27-34, Doi : 10.1021/nn3043572

[GASPA 2004] S. Gasparini, M. Migliore and J. C. Magee, 2004, On the Initiation and Propagation of Dendritic Spikes in CA1 Pyramidal Neurons, Journal of Neuroscience, 24 :11046-11056, Doi : 10.1523/JNEUROSCI.2520-04.2004

[GEOR 2013] S. George, J. Hasler, S. Koziol, S. Nease and S. Ramakrishnan, 2013, Low Power Dendritic Computation for Wordspotting, Journal of Low Power Electronics and Applications, 3 :73-98, Doi : 10.3390/jlpea3020073

[GIBS 1999] J. R. Gibson, M. Beierlein and B. W. Connors, 1999, Two Networks of electrically coupled inhibitory neurons in neocortex, Nature, 402 :75-79, Doi : 10.1038/47035

[GRASS 2011] F. Grassia, L. Buhry, T. Levi, J. Tomas, A. Destexhe and S. Saïghi, 2011, Tunable neuromimetic integrated system for emulating cortical neuron models, Frontiers in Neuroscience, 5 :134, Doi : 10.3389/fnins.2011.00134

[GRASS 2016] F. Grassia, T. Kohno and T. Levi, 2016, Digital hardware implementation of a stochastic two-dimensional neuron model, Journal of Physiology-Paris, 110 :409-416, Doi : 10.1016/j.jphysparis.2017.02.002

[GRESC 2011] M. Greschner, J. Shlens, C. Bakolitsa, G. D. Field, J. L. Gauthier, L. H. Jepson, A. Sher, A. M. Litke and E. J. Chichilnisky, 2011, Correlated firing among major ganglion cell types in primate retina, The Journal of Physiology, 589 :75-86, Doi : 10.1113/jphysiol.2010.193888

[GRIEN 2012] C. Grienberger and A. Konnerth, 2012, Imaging Calcium in Neurons, Cell Press, Neuron, 73 :862-885, Doi : 10.1016/j.neuron.2012.02.011

[GUTN 1990] B. W. Connors and M. J. Gutnick, 1990, Intrinsic firing patterns of diverse neocortical neurons, Cell Press, Trends in Neurosciences, 13 :99-104, Doi : 10.1016/0166-2236(90)90185-D

[HAMM 1971] P. L. Hammer and E. Shlifer, 1971, Applications of pseudo-Boolean methods to economic problems, Theory and Decision, 1 :296-308, Doi : 10.1007/BF00139572

[HAMM 2007] C. Hammond, H. Bergman and P. Brown, 2007, Pathological synchronization in Parkinson's disease : networks, models and treatments, Cell Press, Trends in Neurosciences, 30 :357-364, Doi : 10.1016/j.tins.2007.05.004

[HASL 2007] P. Hasler, S. Kozoil, E. Farquhar and A. Basu, 2007, Transistor channel dendrites implementing HMM classifiers, IEEE, International Symposium on Circuits and Systems (ISCAS), Doi : 10.1109/ISCAS.2007.378287

[HAY 2011] E. Hayn, S. Hill, F. Schurmann, H. Markram and I. Segev, 2011, Models of Neocortical Layer 5b Pyramidal Cells Capturing a Wide Range of Dendritic and Perisomatic Active Properties, PLOS Computational biology , 7 :7, Doi : 10.1371/journal.pcbi.1002107

[HEBB 1949] D. O. Hebb, 1949, Organization of Behavior,ISBN : 978-0805843002

[HELE 2006] H. C. Lai and L. Y. Jan, 2006, The distribution and targeting of neuronal voltage-gated ion channels, Nature Reviews Neuroscience, 7 :548-562, Doi : 10.1038/nrn1938

[HILL 2001] A. A. V. Hill, J. Lu, M. A. Masino, O. H. Olsen and R. L. Calabrese, 2001, A Model of a Segmental Oscillator in the Leech Heartbeat Neuronal Network, Journal of Computational Neuroscience, 10 :281-302, Doi : 10.1023/A :1011216131638

[HOCH 2012] L. R. Hochberg, D. Bacher, B. Jarosiewicz, N. Y. Masse, J. D. Simeral, J. Vogel, S. Haddadin, J. Liu, S. S. Cash, P. Van Der Smagt and J. P. Donoghue, 2012, Reach and grasp by people with tetraplegia using a neurally controlled robotic arm, Nature, 485 :372-375, Doi : 10.1038/nature11076

[HODG 1936] A.L. Hodgkin and W. A. H. Rushton, 1936, The electrical constants of a crustacean nerve fibre, Royal Society, 133 :444, Doi : 10.1098/rspb.1946.0024

[HODG 1948] A. L. Hodgkin, 1948, The local electric changes associated with repetitive action in a non-medullated axon, The Journal of physiology, 107 :165-181, Doi : 10.1113/jphysiol.1948.sp004260

[HODG 1952] A. L. Hodgkin and A. F. Huxley, 1952, A Quantitative description of membrane current and its application to conduction and excitation in nerve, Society of Mathematical Biology, 117 :500-544, Doi : 10.1113/jphysiol.1952.sp004764

[HU 2016] M. Hu, J. P. Strachan, Z. Li, E. M. Grafals, N. Davila, C. Graves, S. Lam, N. Ge, J. J. Yang and R. S. Wiliams, 2016, Dot-product engine for neuromorphic computing : programming 1T1M crossbar to accelerate matrix-vector multiplication, IEEE, 53nd ACM/EDAC/IEEE Design Automation Conference (DAC), Doi : 10.1145/2897937.2898010

[HUKU 1996] K. Hukushima and K. Nemoto, 1996, Exchange Monte Carlo Method and Application to Spin Glass Simulations, Journal of the Physical Society of Japan, 65 :1604-1608, Doi : 10.1143/JPSJ.65.1604

[HUYS 2006] Q. J. M. Huys, M. B. Ahrens and L. Paninski, 2006, Efficient Estimation of Detailed Single-Neuron Models, Journal of Neurophysiology, 96 :872-890, Doi : 10.1152/jn.00079.2006

[IDE 2017] J. S. Ide, F. A. Cappabianco, F. A. Faria and C-S. R. Li, 2017, Detrended Partial Cross Correlation for Brain Anaysis, Advances in Neural Information Processing Systems, 31st congerence on Neural Information Processing Systems (NIPS 2017)

[INDI 2007] G. Indiveri and S. Fusi, 2007, Spike-based learning in VLSI networks of integrate-and-fire neurons, IEEE, International Symposium on Circuits and Systems (ISCAS), Doi : 10.1109/ISCAS.2007.378290

[IZHI 2000] E. M. Izhikevich, 2000, Dynamical Systems in Neuroscience, Computational Neuroscience Series, -, Doi : ISBN : 9780262090438

[IZHI 2003] E. M. Izhikevich, 2003, Simple Model of Spiking Neurons, IEEE Transactions on Neural Networks, 14 :1569-1572, Doi : 10.1109/TNN.2003.820440

[IZHI 2004] E. M. Izhikevich, 2004, Which Model to Use fo Cortical Spiking Neurons, IEEE Transactions on Neural Networks, 15 :1063-1070, Doi : 10.1109/TNN.2004.832719

[JACK 2016] A. Jackson, 2016, Neural interfaces take another step forward, Nature, 539 :177-178, Doi : 10.1038/539177a

[JORD 2018] J. Jordan, T. Ippen, M. Helias, I. Kitayama, M. Sato, J. Igarashi, M. Diesmann and S. Kunkel, 2018, Extremely Scalable Spiking Neuronal Network Simulation Code : From Laptops to Exascale Computers, Frontiers in Neuroinformatics, 12 :2, Doi : 10.3389/fninf.2018.00002

[JOUC 2016] S. Joucla, M. Ambroise, T. Levi, T. Lafon, P. Chauvand, S. Saï ghi, Y. Bornat, N. Lewis, S. Renaud and B. Yvert, 2016, Generation of Locomotor-Like Activity in the Isolated Rat Spinal Cord Using Intraspinal Electrical Microstimulation Driven by a Digital Neuromorphic CPG, Frontiers in Neuroscience, 10 :67, Doi : 10.3389/fnins.2016.00067

[KANDE 1960] W. A. Spencer and E. R. Kandel, 1960, Electrophysiology of Hippocampal Neurons : IV. Fast Prepotentials, Journal of Neurophysiology, 24 :272-285, Doi : 10.1152/jn.1961.24.3.272

[KAWA 2017] J. Kawada, S. Kaneda, T. Kirihara, A. Maroof, T. Levi, L. Eggan, T. Fujii and Y. Ikeuchi, 2017, Generation of a Motor Nerve Organoid with Human Stem Cell-Derived Neurons, Cell Press, Stem Cell Reports, 9 :1441-1449, Doi : 10.1016/j.stemcr.2017.09.021

[KEND 1981] E. Kandel, T. Jessell and J. H. Schwartz, 1981, Principles of Neuroscience, Elsevier, ISBN : 9780071390118

[KERN 1981] D. Kernell and B. Zwaagstra, 1981, Input Conductance, Axonal conduction velocity and cell size among hindlimb motoneurones of the cat , Brain Research,

204 :311-326, Doi : 10.1016/0006-8993(81)90591-6

[KESS 2010] K. C. Ess, 2010, Tuberous Sclerosis Complex : A Brave New World, National Institue of Health (NIH), 23 :189-193, Doi : 10.1097/WCO.0b013e32832c4ff5

[KHOYa 2019] F. Khoyratee, F. Grassia, S. Saïghi and T. Levi, 2019, Optimized Real-Time Biomimetic Neural Network on FPGA for Bio-Hybridization, Frontiers in Neuroscience, 13 :377, Doi : 10.3389/fnins.2019.00377

[KHOYb 2019] F. Khoyratee, S. M. Nishikawa, L. Zhongyue, S. H. Kim, S. Saïghi, T. Fujii, Y. Ikeuchi, K. Aihara and T. Levi, 2019, Biomimetic Spiking Neural Network (SNN) Systems for 'In Vitro' Cells Stimulation, IEEE, IEEE International Symposium on Circuits and Systems (ISCAS), Doi : 10.1109/ISCAS.2019.8702407

[KING 2018] A. D. King, W. Bernoudy, J. King, A. J. Berkley and T. Lanting, 2018, Emulating the coherent Ising machine with a mean-field algorithm, arXiv, -, Doi : 1806 :08422

[KIRI 2019] T. Kirihara, Z. Luo, S. Y. A. Chow, R. Misawa, J. Kawada, S. Shibata, F. Khoyratee, C-A. Vollette, V. Voltz, T. Levi, T. Fujii and Y. Ikeuchi, 2019, A Human Induced Pluripotent Stem Cell-Derived Tissue Model of a Cerebral Tract Connecting Two Cortical Regions, iScience, 14 :301-311, Doi : 10.1016/j.isci.2019.03.012

[KJAE 1996] O. Kjaerulff and O. Kiehn, 1996, Distribution of Networks Generating and Coordinating Locomotor Activity in the Neonatal Rat Spinal Cord In Vitro : A Lesion Study, Journal of Neuroscience, 16 :5777-5794, Doi : 10.1523/JNEUROSCI.16-18-05777.1996

[KOCH 2014] G. Kochenberger, J-K. Hao, F. Glover, M. Lewis, Z. Lü, H. Wang and Y. Wang, 2014, The unconstrained binary quadratic programming problem : a survey, Journal of Combinatorial Optimization, 28 :58-81, Doi : 10.1007/s10878-014-9734-0

[KOHN 2016] T. Kohno, M. Sekikawa, J. Li, T. Nanami and K. Aihara, 2016, Qualitative-Modeling-Based Silicon Neurons and Their Networks, Frontiers in Neuroscience, 10 :273, Doi : 10.3389/fnins.2016.00273

[KRAR 1978] J. Krarup and P. M. Pruzan, 1978, Computer-aided layout design, Mathematical Programming in Use, 9 :75-94, Doi : 10.1007/BFb0120827

[KUNG 1978] H. T. Kung and C. E. Leiserson, 1978, Systolic Arrays for VLSI, Sparse Matrix Proceedings, , Doi :

[LAUGH 1970] D. J. Laughhunn, 1970, Quadratic Binary Programming with Application to Capital-Budgeting Problems, Operations Research, 18 :375-565, Doi : 10.1287/opre.18.3.454

[LEBED 2009] M. A. L. Nicolelis and M. A. Lebedev, 2009, Principles of neural ensemble physiology underlying the operation of brain–machine interfaces, Nature Reviews Neuroscience, 10 :530-540, Doi : 10.1038/nrn2653

[LELEU 2019] T. Leleu, Y. Yamamoto, P. L. McMahon and K. Aihara, 2019, Destabiliza-

tion of Local Minima in Analog Spin Systems by Correction of Amplitude Heterogeneity, Physical Review Letters, 122 :1, Doi : 10.1103/PhysRevLett.122.040607

[LEMA 2014] G. Le Masson, S. Przedborski and L.F. Abbott, 2014, A Computational Model of Motor Neuron Degeneration, Cell Press, Neuron, 83 :975-988, Doi : 10.1016/j.neuron.2014.07.001

[LEVI 1968] H. Levitan, J. P. Segundo and G. P. Moore, 1968, Statistical Analysis of Membrane Potential Fluctuations, Biophysical Journal, 8(11) :1256-1274, Doi : 10.1016/S0006-3495(68)86554-3

[LEVI 2017] T. Levi, F. Khoyratee, S. Saïghi and Yoshiho Ikeuchi, 2017, Digital implementation of Hodgkin-Huxley neuron model for neurological diseases studies, Artificial Life and Robotics, 23 :10–14, Doi : 10.1007/s10015-017-0397-7

[LEVI 2018] T. Levi, P. Bonifazi, P. Massobrio and M. Chiappalone, 2018, Closed-Loop Systems for Next-Generation Neuroprostheses, Frontiers in Neuroscience, 12 :26, Doi : 10.3389/fnins.2018.00026

[LIU 2004] S. C. Liu and R. Douglas, 2004, Temporal Coding in a Silicon Network of Integrate-and-Fire Neurons, IEEE Transactions on Neural Networks, 15 :1305-1314, Doi : 10.1109/TNN.2004.832725

[LONDO 2010] M. London, A. Roth, L. Beeren, M. Hausser and P. E. Latham, 2010, Sensitivity to perturbations in vivo implies high noise and suggests rate coding in cortex, Nature, 466 :123-127, Doi : 10.1038/nature09086

[MAIN 1995] Z. F. Mainen and T. J. Sejnowski, 1995, Reliability of Spike Timing in Neocortical Neurons, Science, 268 :1503-1506, Doi : 10.1126/science.7770778

[MAIN 1996] Z. F. Mainen and T. J. Sejnowski, 1996, Influence of dendritic structure on firing pattern in model neocortical neurons, Nature, 382 :363-366, Doi : 10.1038/382363a0

[MAIN 1998] Z. F. Mainen and T. J. Sejnowski, 1998, Modeling Active Dendritic Processes in Pyramidal Neurons, MIT Press, Methods in Neuronal Modeling

[MAKI 2018] T. Maki-Marttunen, G. Halmes, A. Devor, C. Metzner, A. M. Dale, O. A. Andreassen and G. T. Einevoll, 2018, A stepwise neuron model fitting procedure designed for recordingswith high spatial resolution : Application to layer 5 pyramidal cellsTuomo, Journal of Neuroscience Methods, 293 :264-283, Doi : 10.1016/j.jneumeth.2017.10.007

[MANW 1999] A. Manwani and C. Koch, 1999, Detecting and estimating signals in noisy cable structures, MIT Press, Neural Computation, 11 :1797-1829, Doi : 10.1162/089976699300015972

[MARD 2001] E. Marder and D. Bucher, 2001, Central pattern generators and the control of rhythmic movements, Cell Press, Current Biology, 11 :986-996, Doi : 10.1016/S0960-9822(01)00581-4

[MARK 2015] H. Markram, E. Muller, S. Ramaswamy, M. W. Reimann, M. Abdellah, C. A. Sanchez, A. Ailamaki, L. Alonso-Nanclares, N. Antille, S. Arsever, G. A. A. Kahou, T. K. Berger, A. Bilgili, N. Buncic, A. Chalimourda, G. Chindemi, J-D. Courcol, F. Delalondre, V. Delattre, S. Druckmann, R. Dumusc, J. Dynes, S. Eilemann, E. Gal, M. E. Gevaert, J-P. Ghobril, A. Gidon, J. W. Graham, A. Gupta, V. Haenel, E. Hay, T. Heinis, J. B. Hernando, M. Hines, L. Kanari, D. Keller, J. Kenyon, G. Khazen, Y. Kim, J. G. King, Z. Kisvarday, P. Kumbhar, S. Lasserre, J-V. Le Bé, B. R.C. Magalhães, A. Merchán-Pérez, J. Meystre, B. R. Morrice, J. Muller, A. Muñoz-Céspedes, S. Muralidhar, K. Muthurasa, D. Nachbaur, T. H. Newton, M. Nolte, A. Ovcharenko, J. Palacios, L. Pastor, R. Perin, R. Ranjan, I. Riachi, J-R. Rodríguez, J. L. Riquelme, C. Rössert, K. Sfyrakis, Y. S., J. C. Shillcock, G. Silberberg, R. Silva, F. Tauheed, M. Telefont, M. Toledo-Rodriguez, T. Tränkler, W. Van Geit, J. V. Díaz, R. Walker, Y. Wang, S. M. Zaninandta, J. DeFelipe, S. L. Hill, I. Segev and F. Schurmann, 2015, Reconstruction and Simulation of Neocortical Microcircuitry, Cell Press, Cell, 163 :277-280, Doi : 10.1016/j.cell.2015.09.029

[MART 2013] E. Martin, W. Cazenave, D. Cattaert and P. Branchereau, 2013, Embryonic alteration of motoneuronal morphology induces hyperexcitability in the mouse model of amyotrophic lateral sclerosis, Neurobiology of Disease , 54 :116-126, Doi : 10.1016/j.nbd.2013.02.011

[MCBA 2001] C. J. McBain and A. Fisahn, 2001, Interneurons Unbound, Nature Reviews Neuroscience, 2 :11-23, Doi : 10.1038/35049047

[MCS 2019] Multichannel Systems, 2019, Stimulus Generator 4002 Manual, Doi : www.multichannelsystems.com

[MELH 1988] R. Melhem, 1988, Parallel solution of linear systems with striped sparse matrices, Parallel Computing, 6 :165-184, Doi : 10.1016/0167-8191(88)90082-8

[MELW 1994] B. W. Mel, 1994, Information Processing in Dendritic Trees, MIT Press, Neural Computation, 6 :1031-1085, Doi : 10.1162/neco.1994.6.6.1031

[MERO 2014] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura, B. Brezzo, I. Vo, S. K. Esser, R. Appuswamy, B. Taba, A. Amir, M. D. Flickner, W. P. Risk, R. Manohar and D. S. Modha, 2014, A million spiking-neuron integrated circuit with a scalable communication network and interface, Science, 345 :668-673, Doi : 10.1126/science.1254642

[MICH 1998] M. Zoli, C. Torri, R. Ferrari, A. Jansson, I. Zini, K. Fuxe and L. F. Agnati, 1998, The emergence of the volume transmission concept, Brain research reviews , 26- :136-147, Doi : 10.1016/S0165-0173(97)00048-9

[MONC 1969] V. B. Mountcastle, W. H. Talbot, H. Sakata and J. Hyvarinen, 1969, Cortical neuronal mechanisms in flutter-vibration studied in unanesthetized monkeys. Neuronal periodicity and frequency discrimination, Journal of Neurophysiology, 32 :452-484, Doi : 10.1152/jn.1969.32.3.452

[MONTE 2012] M. Monteforte and F. Wolf, 2012, Dynamic Flux Tubes Form Reservoirs of Stability in Neuronal Circuits, Physical Review X, 2 :12, Doi : 10.1103/PhysRevX.2.041007

[NATA 2018] A. Natarajan and J. Hasler, 2018, Hodgkin-Huxley Neuron and FPAA Dynamics, IEEE Transactions on Biomedical Circuits and Systems, 12 :918-926, Doi : 10.1109/TBCAS.2018.2837055

[NAUN 2006] B. Naundorf, F. Wolf and M. Volgushev, 2006, Unique features of action potential initiation in cortical neurons, Nature, 440 :1060-1063, Doi : 10.1038/nature04610

[NAWR 2007] M. P. Nawrot, C. Boucsein, V. Rodriguez-Molina, A. Aertsen, S. Grün and S. Rotter, 2007, Serial interval statistics of spontaneous activity in cortical neurons in vivo and in vitro, Neurocomputing, 70 :1717-1722, Doi : 10.1016/j.neucom.2006.10.101

[NAZA 2015] S. Nazari, K. Faez, M. Amiri and E. Karami, 2015, A digital implementation of neuron-astrocyte interaction for neuromorphic applications, Neural Networks, 66 :79-90, Doi : 10.1016/j.neunet.2015.01.005

[NISHI 2018] S. M. Nishikiwa, S. H. Kim, Z. Luo, T. Kirihara, Y. Ikeuchi and T. Fujii, 2018, Biomimetic microfluidic neurons for bio-hybrid experiments, Artificial Life and Robotics, 23 :402-408, Doi : 10.1007/s10015-018-0452-z

[NISHI 2019] S. M. Nishikawa, F. Khoyratee, A-C. Eiler, S. H. Kim, S. Ihida, H. Toshiyoshi, A. Tixier-Mita, Y. Ikeuchi, T. Kohno, K. Aihara, T. Fuji and T. Levi, 2019, Neuro-hybrid systems : the future for biomedical applications, Seisan Kenkyu, 71 :787-790, Doi : https ://doi.org/10.11188/seisankenkyu.71.787

[OBER 1995] S. F. Oberman and M. J. Flynn, 1995, An Analysis of Division Algorithms and Implementations, IEEE Transaction on Computers, 46 :833-854, Doi : 10.1109/12.609274

[ODOH 2011] J. E. OĎoherty, M. A. Lebedev, P. J. Ifft, K. Z. Zhuang, S. Shokur, H. Bleuler and M. A. L. Nicolelis, 2011, Active tactile exploration enabled by a brain-machine-brain interface, Nature, 479 :228-231, Doi : 10.1038/nature10489

[OLLE 2006] O. Lindvall and Z. Kokaia, 2006, Stem cells for the treatment of neurological disorders, Nature, 441 :1094-1096, Doi : 10.1038/nature04960

[OSOR 2016] R. R. Osorio, 2016, Pipelined FPGA Implementation of numerical Integration of the HH Model, IEEE , 27th International Conference on Application-specific Systems, Architectures and Processors (ASAP), Doi : 10.1109/ASAP.2016.7760794

[PATI 1998] M. M. Patil, C. Linster, E. Lubenov and M. E. Hasselmo, 1998, Cholinergic Agonist Carbachol Enables Associative Long-Term Potentiation in Piriform Cortex Slices, The American Physiological Society, 80 :2467-2474, Doi : 10.1152/jn.1998.80.5.2467

[PELV 2008] D. P. Pelvig, H. Pakkenberg, A. K. Stark and B. Pakkenberg, 2008, Neocortical glial cell numbers in human brains, Neurobiology of Aging, 29 :1754-1762, Doi : 10.1016/j.neurobiolaging.2007.04.013

[POLY 1964] B. T. Polyak, 1964, Some methods of speeding up the convergence of iteration methods, USSR Computational Mathematics and Mathematical Physics, 4 :1-17, Doi : 10.1016/0041-5553(64)90137-5

[POSP 2008] M. Pospischil, M. Toledo-Rodriguez, C. Monier, Z. Piwkowska, T. Bal, Y. Frégnac, H. Markram and A. Destexhe, 2008, Minimal Hodgkin-Huxley type model for different classes of cortical and thalamic neurons, Biological Cybernetics , 99 :427–441, Doi : 10.1007/s00422-008-0263-8

[POTT 2014] S. M. Potter, A. El Hady and E. E. Fetz, 2014, Closed-loop neuroscience and neuroengineering, Frontiers in Neural Circuits, 8 :115, Doi : 10.3389/fncir.2014.00115

[PRAS 2007] V. K. Prasanna and G. R. Morris, 2007, Sparse matrix computations on reconfigurable, IEEE, 40 :58-64, Doi : 10.1109/MC.2007.103

[PRES 2014] S. A. Prescott, 2014, Excitability : Types I, II, and III, Encyclopedia of Computational Neuroscience, Doi : 10.1007/978-1-4614-6675-8_ 151

[PUJO 2000] J. Pujol, J. Bello, J. Deus, N. Cardoner, J. L. Marti-Vilalta and A. Capdevila, 2000, Beck Depression Inventory factors related to demyelinating lesions of the left arcuate fasciculus region, Neuroimaging, 99 :151-159, Doi : 10.1016/S0925-4927(00)00061-5

[QIAO 2015] N. Qiao, H. Mostafa, F. Corradi, M. Osswald, F. Steganini, D. Sumislawska and G. Indiveri, 2015, A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128K synapses, Frontiers in Neuroscience, 9 :141, Doi : 10.3389/fnins.2015.00141

[RALL 1959] W. Rall, 1959, Branching dendritic trees and motoneuron membrane resistivity, Experimental Neurology, 1 :491-527, Doi : 10.1016/0014-4886(59)90046-9

[RAO 1971] M. R. Rao, 1971, Cluster Analysis and Mathematical Programming, Journal of the American Statistical Association, 66 :622-626, Doi : 10.1080/01621459.1971.10482319

[RASAM 2019] M. Rasamuel, D. Gaffé, T. Levi and B. Miramond, 2019, Synchronous Approach for Modeling Spiking Neurons, IEEE, Biomedical Circuits and Systems Conférence (BIOCAS)

[RECC 1999] M. Recce, 1999, Encoding information in neuronal activity, Pulsed neural network, 111-131, Doi : -

[RENA 2007] S. Renaud, J. Tomas, Y. Bornat, A. Daouzli and S. Saïghi, 2007, Neuromimetic ICs with analog cores an alternative for simulating spiking neural networks, IEEE, International Symposium on Circuits and Systems, Doi : 10.1109/ISCAS.2007.378286

[RENA 2014] A. Renart and C. K. Machens, 2014, Variability in neural activity and behavior, Current Opinion in Neurobiology, 25 :211-220, Doi : 10.1016/j.conb.2014.02.013

[RHYS 1970] J. M. W. Rhys, 1970, A selection problem of shared fixed costs and network flows, Management Science, 17 :127-246, Doi : 10.1287/mnsc.17.3.200

[RINZ 1995] V. Booth and J. Rinzel, 1995, A minimal, compartmental model for a dendritic origin of bistability of motoneuron firing patterns, Journal of Computational Neuroscience, 2 :299-312, Doi : 10.1007/BF00961442

[ROBE 1958] J. E. Robertson, 1958, A new Class of Digital Division Methods, IEEE, EC-7 :218-222, Doi : 10.1109/TEC.1958.5222579

[SAIG 2011] S. Saïghi, Y. Bornat, J. Tomas, G. Le Masson and S. Renaud, 2011, A Library of Analog Operators Based on the Hodgkin-Huxley Formalism for the Design of Tunable, Real-Time, Silicon Neurons, IEEE, 5 :3-19, Doi : 10.1109/TBCAS.2010.2078816

[SCHE 2007] J. Schemmel, D. Bruderle, K. Meier and B. Ostendorg, 2007, Modeling Synaptic Plasticity within Networks of Highly Accelerated I& F Neurons, IEEE , International Symposium on Circuits and Systems (ISCAS), Doi : 10.1109/ISCAS.2007.378289

[SCHIL 2000] J. Schiller, G. Major, H. J. Koester and Y. Schiller, 2000, NMDA spikes in basal dendrites of cortical pyramidal neurons, Nature, 404 :285-289, Doi : 10.1038/35005094

[SHEVE 1995] I. A. Shevelev, R. V. Novikova, N. A. Lazareva, A. S. Tikhomirov and G. A. Sharaev, 1995, Sensitibity to cross-like figures in the cat striate neurons, Neuroscience, Neuroscience, 69 :51-57, Doi : 10.1016/0306-4522(95)00188-O

[SHIGE 2003] S. Shinomoto, K. Shima and J. Tanji, 2003, Differences in Spiking Patterns Among Cortical Neurons, Neural Computation, 15 :2823-2842, Doi : 10.1162/089976603322518759

[SIMO 2007] M. F. Simoni and S. P. DeWeerth, 2007, Sensory Feedback in a Half-Center Oscillator Model, IEEE Transactions on Biomedical Engineering, 54 :193-204, Doi : 10.1109/TBME.2006.886868

[SORE 2004] M. Sorensen, S. DeWeerth, G. Cymbalyuk and R. L. Calabrese, 2004, Using a Hybrid Neural System to Reveal Regulation of Neuronal Network Activity by an Intrinsic Current, The Journal of Neuroscience, 24 :5427-5438, Doi : 10.1523/JNEUROSCI.4449-03.2004

[SPIL 2016] J. Spillane, D. M. Kullmann and M. G. Hanna, 2016, Genetic neurological channelopathies - molecular genetics and clinical phenotyoes, Journal of Neurology, Neurosurgery, Psychiatry, 87 :37-48, Doi : 10.1136/jnnp-2015-311233

[SUZA 2009] S. Herculano-Houzel, 2009, The human brain in numbers : a linearly scaled up primate brain, Frontiers in Human Neuroscience, 3 :31, Doi : 10.3389/neuro.09.031.2009

[SWAN 2017] L. W. Swanson, J. D. Hahn and O. Sporns, 2017, Organizing principles for the cerebral cortex network of commissural and association connections, PNAS , 114(45) :E9692-E9701, Doi : 10.1073/pnas.1712928114

[THAN 2019] S. Le Thanh, N. Lobato-Dauzier, F. Khoyratee, R. Beaubois, T. Fujii, A. Genot and T. Levi, 2019, Low power and massively parallel simulation of oscillatory biochemical networks on FPGA, IEEE, 2019 IEEE Biomedical Circuits and Systems Conference (BioCAS), Doi : 10.1109/BIOCAS.2019.8919020

[THOM 1904] W. Thomson, 1904, Molecular Dynamics and the Wave Theory of Light, Cambridge University Press, -, Doi : ISBN : 9781108007672

[THOR 1996] S. Thorpe, D. Fize and C. Marlot, 1996, Speed of processing in the human visual system, Nature, 381 :520-522, Doi : 10.1038/381520a0

[TIXIER 2019] A. Tixier-Mita, S. Ihida, D. Blanchard, M. Shinohara, A-C. Eiler, G. A. Cathcart, P-M. Faure, T. Kohno, Y. Sakai, T. Levi and H. Toshiyoshi, 2019, 2D Dielectrophoresis using an active matrix array made by thin-film transistor technology, IEEJ Transactions on Electrical and Electronic Engineering, 14 :9, Doi : https ://doi.org/10.1002/tee.22979

[TOLE 2004] M. Toledo-Rodriguezn B. Blulenfeld, C. Wu, J. Luo, B. Attali, P. Goodman and H. Markram, 2004, Correlation Maps Allow Neuronal Electrical Properties to be Predicted from Single-cell Gene Expression Profiles in Rat Neocortex, Cerebral Cortex, 14 :1310–1327, Doi : 10.1093/cercor/bhh092

[TOMI 2001] M. Tomita, 2001, Whole-cell simulation : a grand challenge of the 21st century, Cell Press, Trends in Biotechnology, 19 :205-210, Doi : 10.1016/S0167-7799(01)01636-5

[TUCK 1988] H. C. Tuckwell, F. Y. M. Wan and J-P. Rospars, 2002, A spatial stochastic neuronal model with Ornstein–Uhlenbeck input current, Biological Cybernetics , 86 :137-145, Doi : 10.1007/s004220100283

[VELL 2008] M. Velliste, S. Perel, M. C. Spalding, A. S. Whitford and A. B. Schwartz, 2003, Cortical control of a prosthetic arm for self-feeding, Nature, 453 :1098-1101, Doi : 10.1038/nature06996

[VOGEL 2004] R. J. Vogelstein, U. Mallik, G. Cauwenberghs, 2004, Silicon spike-based synaptic array and address-event transceiver, IEEE, International Symposium on Circuits and Systems (ISCAS), Doi : 10.1109/ISCAS.2004.1329585

[VOLD 1959] J. E. Volder, 1959, The CORDIC Trigonometric Computing Technique, IEEE, IRE Transactions on Electronic Computers, EC-8 :330-334, Doi : 10.1109/TEC.1959.5222693

[WANG 2013] R. Wang, G. Cohen, K. M. Stiefel, T. J. Hamilton, J. Tapson and A. V. Schaik, 2013, An FPGA implementation of a polychronous spiking neural network with delay adaptation, Frontiers in Neuroscience, 7 :14, Doi : 10.3389/fnins.2013.00014

[WEBB 1976] A. C. Webb and B. D. Burns, 1976, The effects of changing levels of arousal on the spontaneous activity of cortical neurones I. Sleep and wakefulness, The Royal Society Publishing, Biological Sciences, 194, Doi : 10.1098/rspb.1976.0075

[WILI 1988] W. Williams and K. Herrup, 1988, The Control of Neuron Number, Annual Review of Neuroscience, 11 :423-453, Doi : 10.1146/annurev.ne.11.030188.002231

[WITZ 1975] C. Witzgall, 1975, Mathematical methods of site selection for Electronic Message Systems (EMS), Final Report National Bureau of Standards, Washington, DC. Operations Research Section, Technical Report, Doi : 10.6028/nbs.ir.75-737

[XILI 2019] Xilinx, 2019, Ultrascale Architecture DSP slice user guide, Xilinx technical document, UG579, Doi : -

[YAMA 2006] K. Takahashi and S. Yamanaka, 2006, Induction of Pluripotent Stem Cells from Mouse Embryonic and Adult Fibroblast Cultures by Defined Factors, Cell Press, Cell, 126 :663-676, Doi : 10.1016/j.cell.2006.07.024

[YEH 2018] F-C. Yeh, S. Panesar, D. Fernandes, A. Meola, M. Yoshino, J. C. Fernandez-Miranda, J. M. Vandtel and T. Verstynen, 2018, Population-averaged atlas of the macroscale human structural connectome and its network topology, NeuroImage, 178 :57-68, Doi : 10.1016/j.neuroimage.2018.05.027

[ZING 2014] B. Zingg, H. Hintiryan, L. Gou, M. Y. Song, M. Bay, M. S. Bienkowski, N. N. Foster, S. Yamashita, I. Bowman, A. W. Toga and H-W. Dong, 2014, Neural Networks of the Mouse Neocortex, Cell Press, Cell, 156 :1096-1111, Doi : 10.1016/j.cell.2014.02.023

Publications de l'auteur

Journaux international

F. Khoyratee, F. Grassia, S. Saïghi and T. Levi, 2019, Optimized Real-Time Biomimetic Neural Network on FPGA for Bio-Hybridization, Frontiers in Neuroscience, 13 :377, Doi : 10.3389/fnins.2019.00377

S. Nishikawa, F. Khoyratee, A-C. Eiler, S. H. Kim, S. Ihida, S. Toshiyoshi, A. Tixier-Mita, Y. Ikeuchi, T. Kohno, K. Aihara, T. Fujii and T. Levi, 2019, Neuro-hybrid systems : the future for biomedical applications, Seisan Kenkyu, 71 :787-790, https ://doi.org/10.11188/seisankenkyu.71.787

T. Kirihara, Z. Luo, S. Y. A. Chow, R. Misawa, J. Kawada, S. Shibata, F. Khoyratee, C-A. Vollette, V. Voltz, T. Levi, T. Fujii and Y. Ikeuchi, 2019, A Human Induced Pluripotent Stem Cell-Derived Tissue Model of a Cerebral Tract Connecting Two Cortical Regions, iScience, 14 :301-311, Doi : 10.1016/j.isci.2019.03.012

T. Levi, F. Khoyratee, S. Saïghi and Yoshiho Ikeuchi, 2017, Digital implementation of Hodgkin-Huxley neuron model for neurological diseases studies, Artificial Life and Robotics, 23 :10–14, Doi : 10.1007/s10015-017-0397-7

Conférences internationales

S. Le Thanh, N. Lobato-Dauzier, F. Khoyratee, R. Beaubois, T. Fujii, A. Genot and T. Levi, 2019, Low power and massively parallel simulation of oscillatory biochemical networks on FPGA, IEEE, 2019 IEEE Biomedical Circuits and Systems Conference (BioCAS), Doi : 10.1109/BIOCAS.2019.8919020

A-C. Eiler, F. Khoyratee, S. Ihida, H. Toshiyoshi, T. Levi and A. Tixier-Mita, 2019, A Bio-Hybrid Platform using Thin-Film-Transistor Arrays and a Biomimetic Neural Network, 11th International Symposium on Michrochemistry and Microsystems, ISMM

F. Khoyratee, S-M. Nishikawa, Z. Luo, S. O. Kim, S. Saighi, T. Fujii, Y. Ikeuchi, K. Aihara and T. Levi, 2019, Biomimetic Spiking Neural Network (SNN) systems for 'in vitro' cells stimulation, IEEE ISCAS 2019, Sapporo, Doi : 10.1109/ISCAS.2019.8702407

S. Nishikawa, F. Khoyratee, S-O. Kim, Y. Ikeuchi, Y. Aihara, T. Fujii and T. Levi, 2019, Biomimetic spike-timing based ionic micro-stimulation for neuron culture, 24th International Conference on Artificial Life and Robotics, ICAROB 2019

S. Nishikawa, F. Khoyratee, Z. Luo, T. Shiraishi, K. Aihara, Y. Ikeuchi, S-H. Kim, T. Fujii and T. Levi, 2018, Neurohybrid system with Spiking Neural Network and biomimetic ionic

microstimulation, MicroTAS 2018, Taiwan

F. Khoyratee, Z. Luo, C-A. Volette, T. Benneteau, R. Beaubois, C. Lange, K. Aihara, T. Fujii, Y. Ikeuchi and T. Levi, 2018, Neuromimetic stimulation applied on cerebral organoid, ISMM 2018, Busan, South Korea

F. Khoyratee, T. Benneteau, A. Tixier-Mita, S. Saighi and T. Levi, 2018, Biohybrid experiments using tunable real-time biomimetic neural network, ISMM 2018, Busan, South Korea

F. Khoyratee, S. Saighi and T. Levi, 2018, Real-time digital implementation of HH neural network on FPGA : cortical neuron simulation, 23th International Conference on Artificial Life and Robotics, ICAROB 2018, Beppu, Japan

T. Kirihara, J. Kawada, Z. Luo, A. Chow, R. Misawa, F. Khoyratee, T. Levi, T. Fujii and Y. Ikeuchi, 2018, A three-dimensional in vitro model of interconnected cerebral regions, Proceedings of Cheminas, 37 :89

F. Khoyratee, T. Levi and S. Saighi, 2017, FPGA Implementation of the Hodgkin-Huxley Model for Neurological Disease study, The 2nd International Symposium on Neuromorphic, non-linear, Neurofluidic Engineering, ISNNE 2017, Bordeaux, France

T. Levi, F. Khoyratee, S. Saighi and Y. Ikeuchi, 2017, Digital implementation of Hodgkin-Huxley neuron model for neurological diseases studies, 22th International Symposium on Artificial Life and Robotics, Beppu, Japan, 23 :10–14, Doi : https ://doi.org/10.1007/s10015-017-0397-7

Conférences nationales

F. Khoyratee, F. Grassia, S. Saïghi and T. Levi, 2019, Réseaux de neurones artificiels biomimétiques pour de la bio-hybridation, Journée Francophone de la Recherche JFR, Japan

F. Khoyratee, K. Aihara and T. Levi, 2018, Neurobio-Hybrid experiments for brain study, Frontiers of NeuroIntelligence, IRCN, Tokyo, Japan

S. Nishikawa, F. Khoyratee, Z. Luo, S-H. Kim, K. Aihara, Y. Ikeuchi, T. Fujii and T. Levi, 2018, Biohybrid system with spiking neural network and ionic microstimulation microfluidic chip, Joint French Japanese technology and bioengineering against liver disorders, CNRS JSPS Workshop, Tokyo, Japan

F. Khoyratee, S. Saighi and T. Levi, 2017, Implémentation du modèle de Hodgkin-Huxley sur FPGA pour l'étude des maladies neurodégénératives, Journées Francophones de la Recherche JFR 2017, Tokyo, Japan (poster)

F. Khoyratee, T. Levi and S. Saighi, 2017, FPGA Implementation of the Hodgkin-Huxley Model for Neurological Disease study, JJC ICON, Bordeaux (poster)

F. Khoyratee, T. Levi and S. Saighi, 2016 FPGA Implementation of the Hodgkin-Huxley Model for Neurological Disease study, GDR BioComp, Lyon (poster)

Annexes

1. Field Programmable Gate Array (FPGA)

Les FPGA sont des composants provenant de la famille des Programmable Logic Device (PLD). Ils sont, comme tous circuits intégrés, composés de semi-conducteurs. Ces circuits sont organisés par cellules logiques interconnectés (4.14.c) appelé Configurable Logic Block (CLB). Ces cellules et leurs interconnexions (4.14.b) peuvent être configurées pour exécuter des fonctions logiques. Les CLB sont eux même composés de « slices » (4.14.d) qui sont des sous-cellules regroupant des Look-Up Table (LUT), des bascules D ou D Flip-Flops (DFF) en anglais, des multiplexeurs (MUX) et des unités arithmétiques et logiques ou Arithmetic and Logic Unit (ALU) en anglais. Les FPGA sont aujourd'hui composés d'autres circuits et fonctions pré-implémentés et préfabriqués. Ils peuvent posséder par exemple des Random Access Memory (RAM) pour la mémoire, des Digital Signal Processing (DSP) pour les multiplications, des Phased-Locked Loop (PLL) pour la gestion de l'horloge, des interfaces d'entrées sorties et pour certains des microprocesseurs.

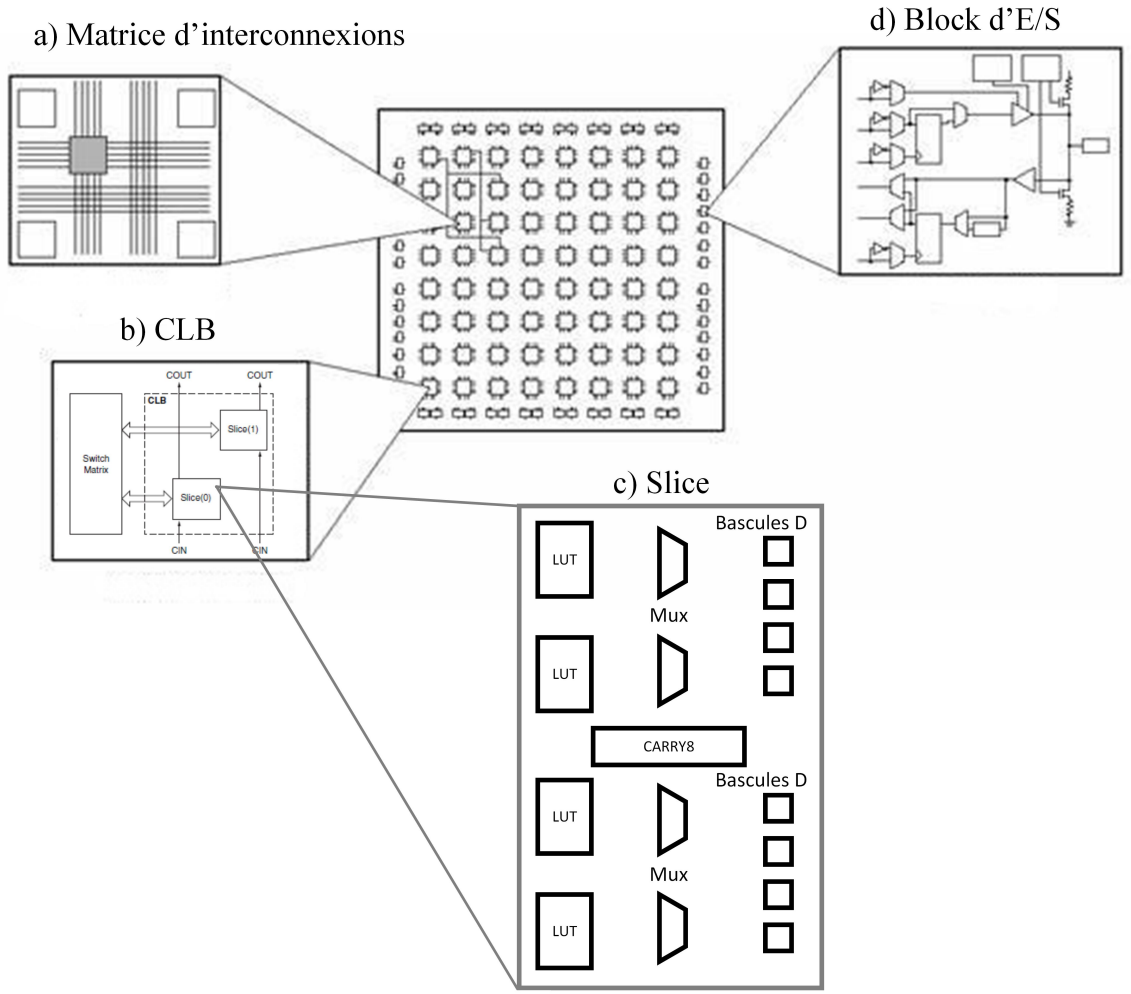

FIGURE 4.14 – Organisation d'un FPGA Xilinx[3]et image du CLB de [XILI 2016] ; a) Block d'entrées et sorties ; b) Matrice d'interconnexions permettant de connecter les différents modules ; c) Configurable Logic Block (CLB) contenant 2 slices ; d) Contenu d'une slice, le nombre de LUT, de Mux et de Bascules ne sont pas représentatives d'une slice Xilinx. Une slice contient également un additioneur/soustracteur 8 entrées.

1.1 Les unités logiques ou Look-Up Table (LUT)

Il a été dit précédemment que les FPGA étaient composés de cellules logiques. Ces derniers sont organisés en LUT et DFF. Ils sont considérés comme les plus petites unités logiques et servent d'éléments quantificateurs pour déterminer la taille d'un design ou la capacité d'un FPGA. Les LUT peuvent fonctionner dans un mode calcul logique et un mode mémoire. La mémoire tout comme les calculs sont limités par les entrées des LUT qui sont généralement entre 2 et 9 pour une sortie. Leurs fonctions est d'effectuer des opérations booléennes (AND, OR, NAND, NOR, NOT, XOR...).

3. https ://www.ni.com/fr-fr/innovations/white-papers/08/fpga-fundamentals.html

1.2 Microprocesseur versus FPGA

II peut être difficile de faire la différence entre un Processeur et un FPGA. Le processeur est un circuit électronique conçu et non modifiable pour exécuter des opérations sur nombres flottants. De nos jours, les processeurs peuvent posséder plusieurs cœurs de calcul. Un cœur de calcul n'effectue qu'une opération à la fois. D'ores et déjà, deux différences peuvent être notées. Tout d'abord, les interconnexions du FPGA peuvent être reconfigurées, ce qui n'est pas le cas du CPU. Aussi, les circuits reconfigurables possèdent l'avantage d'exécuter des fonctions en parallèles. Il est donc possible de faire plusieurs calculs en même temps. Il est quand même bon de noter que la fréquence de fonctionnement des CPU est généralement plus élevée que les FPGA (ordre du GHz pour les CPU, 100 voire 200 MHz pour les FPGA les plus récents). La présence de PLL permet de multiplier les fréquences mais il faudra alors prendre en compte la limite de fonctionnement des composants logiques qui, généralement, ne dépasseront pas les 500MHz concernant les FPGA du commerce.

1.3 Processus de reconfiguration

La reconfiguration hardware des FPGA se fait par un processus particulier. Tout d'abord, il existe des langages de description qui permettent de décrire le design souhaité et les interconnexions entre composants. Ces langages tel que le Very High Speed Integrated Circuit (VHSIC) Hardware Description Language (VHDL) ou le Verilog sont au niveau matériel, à ne pas confondre avec les langages informatiques comme le C, C++ ou Java. La syntaxe peut être similaire mais les fonctions sont différentes. Malgré tout, Il existe des moyens de transformer un script en langage C ou Python en fichier de reconfiguration du FPGA. D'autres utilitaires de plus haut niveau permettent de transformer des schémas blocs en fichier de reconfigurations (Simulink) et d'autres langage comme celui présenté par [RASAM 2019] permet de convertir une écriture formelle en langage de description matérielle. Les travaux précédents de l'équipe sont réalisés en VHDL, langage que nous avons conservé car il offre une certaine flexibilité comparée au verilog. Une fois le/les systèmes numériques réalisés, plusieurs étapes s'offrent au concepteur. Tout d'abord, il y a la simulation, ensuite la synthèse et enfin l'implémentation. La synthèse permet de générer et de reconstituer un schéma logique du design à partir des fichiers de descriptions (HDL). L'implémentation, contrairement à la synthèse prend en compte les contraintes physiques, s'assure d'un placement des composants et d'un routage optimal des connexions des entrées aux sorties. Bien entendu, le logiciel ne place pas de composants mais choisis les composants qui seront utilisés dans le FPGA. Ces étapes sont réalisées par des outils mis en place par les constructeurs. A noter qu'il existe aussi des outils open source qui permettent de le faire. La simulation est une étape permettant au concepteur de tester son système. Elle possède trois niveaux, la simulation comportementale, la simulation post-synthèse et la simulation post-implémentation. Chaque étape rapproche la simulation du réel, Il ne faut cependant pas oublier que le dernier niveau d'implémentation est le plus proche du réel mais n'est pas le réel. Une fois que le logiciel du constructeur finit le placement et le routage des composants, l'étape finale est la construction d'un fichier de reconfiguration qui sera transmis au FPGA lui indiquant la manière dont il doit être configuré.

Pour résumer, le FPGA est une technologie intéressante en le comparant au GPU, au CPU et à l'ASIC. Son accessibilité, sa flexibilité, sa portabilité en font un outil idéal. Si l'on rajoute la possibilité de faire du calcul en temps réel, il s'agit de la meilleure plateforme pour l'application voulue. Le FPGA est organisé en unités de différents niveaux. Les cellules logiques

comprennent des slices qui sont composés de LUT, de bascules et de multiplexeurs. Les LUT étant l'unité quantitative déterminant la taille d'un FPGA, elle permet d'exécuter les opérateurs booléens configurés. Le FPGA possède aussi d'autres circuits électronique permettant d'ajouter des fonctions comme, pour ne citer que ceux-là, la mémoire, la gestion d'horloges ou la gestion des entrées et des sorties. La reconfiguration des interconnexions et des cellules se fait en plusieurs étapes. Elle commence par l'écriture d'un code via un langage de description. S'en suit une étape de conception du circuit numérique, de placement et de routage et enfin, de construction d'un fichier de reconfiguration qui sera envoyé au FPGA pour exécuter les fonctions voulues. Cette partie est nécessaire à la compréhension de la partie relative à la création et à l'optimisation des circuits numériques pour les implémentations des modèles et à leurs simplifications.

2. Temps de propagation et pipeline

2.1 Temps de propagation

Le terme "temps de propagation", lors d'une implémentation matérielle numérique, désigne le temps que met une information à se propager dans les composants. Pour un système synchrone, il est important de prendre en compte ce temps de propagation car, l'information est généralement sauvegardée/prise en compte à un moment précis. Il peut être sur la montée vers un état haut du signal d'horloge (front montant) ou sur la descente (front descendant). La 4.15.a) représente un circuit logique entre deux bascules D. La bascule D retranscrit l'entrée sur la sortie à chaque front montant et agit comme une mémoire car elle conserve l'information précédente. Si le délai de propagation est trop important comme sur la 4.15.b), le résultat du calcul n'est pas sauvegardé. Le temps de propagation concerne donc les composants logiques par lequel passe l'information mais aussi, le routage entre les composants. Ainsi, plus la fréquence d'horloge augmentera et plus le temps de propagation devra être réduit et les connexions entre composants minimisés. Le circuit logique entre deux bascule implique une transmission de données à travers plusieurs composants, on parle alors de chemin de transmission. Dans un système, le chemin de transmission le plus susceptible d'apporter du délai et donc qui contient potentiellement le plus de composants est appelé chemin critique.

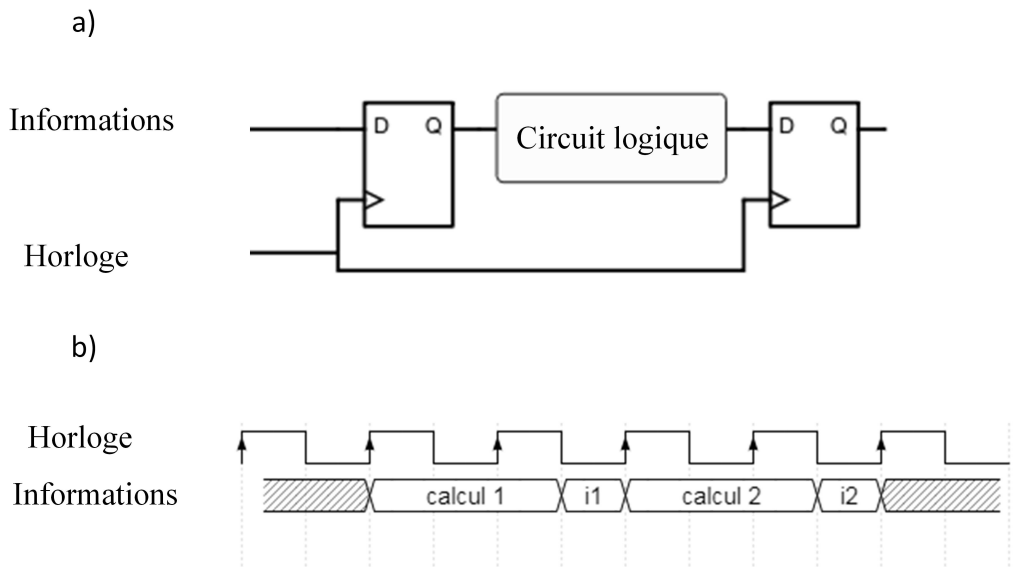

FIGURE 4.15 – Schéma et explication des temps de propagation ; a) circuit logique entre deux bascules D représentant le traitement d'une information et le délai de la propagation de l'information à travers le circuit logique ; b) Chronogramme montrant deux calculs « calcul 1 » et « calcul 2 » dont le circuit induit un temps de propagation assez élevé pour que les informations i1 et i2 soient prise en compte par la bascule

Concernant les bascules D synchronisée à une horloge, il faut aussi prendre en compte le temps de configuration (ou setup time) et le temps de maintien (ou hold time). Ces deux termes sont à prendre en compte lors de la création d'un système numérique car ils déterminent le moment où la bascule prend en compte l'information. La figure 4.16 montre cette caractéristique des bascules. D'une part, le setup time détermine la période pendant laquelle les données doivent être stables pour être prises en compte par le front montant de l'horloge. D'autre part, le hold time détermine le temps minimal, après le front montant de l'horloge, pendant lequel les données transmises à la bascule doivent être stables. Si les données ne respectent pas l'une des deux conditions, il faudra soit diminuer la fréquence d'horloge, soit diminuer le nombre de composants utilisés entre deux bascules. Ces données peuvent être retrouvées dans la documentation technique du composant. Le choix des algorithmes et des composants qui sont présentés dans la suite de ce manuscrit ont été faits en fonction de ce principe.

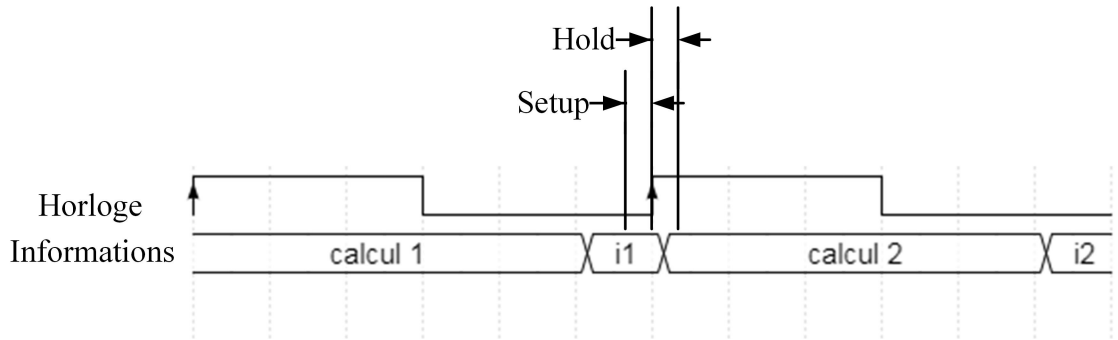

FIGURE 4.16 – Temps de configuration et temps de maintien d'une bascule D sur un chronogramme, le temps setup

2.2 Pipeline

Dans le cas de l'implémentation d'un algorithme itératif, il est possible de calculer plusieurs itérations en un cycle d'horloge. Dépendant de la fréquence de fonctionnement, il est aussi possible de rajouter des étapes appelés pipelines. Un pipeline est une configuration dans lequel les données circulent les unes à la suite des autres dans les différentes unités de calculs. La figure 4.17 montre l'exemple du calcul d'un canal ionique des équations de Hodgkin-Huxley sous forme de pipeline. L'avantage des pipelines est qu'il augmente la vitesse de débit des données et de calculs et permet d'ajouter des étapes à base de bascule D dans le circuit afin de réduire le chemin de transmission et les temps de propagations. En revanche, ce type de configuration empêche le bouclage sur un même composant afin de réutiliser celui-ci dans une même fonction ou le partage d'opérateurs (comme les multiplieurs par exemple) et nécessite des composants à chaque étape de calculs ce qui a pour effet d'augmenter la quantité de ressources. Il est bon de noter que les algorithmes de synthèses, fournis par les constructeurs pour convertir un langage de description matériel en un fichier de configuration pour le FPGA, peuvent modifier le circuit décrit par le concepteur pour partager certaines ressources comme les additionneurs ce qui permet de réduire les ressources totales au prix de temps de propagation plus élevé.

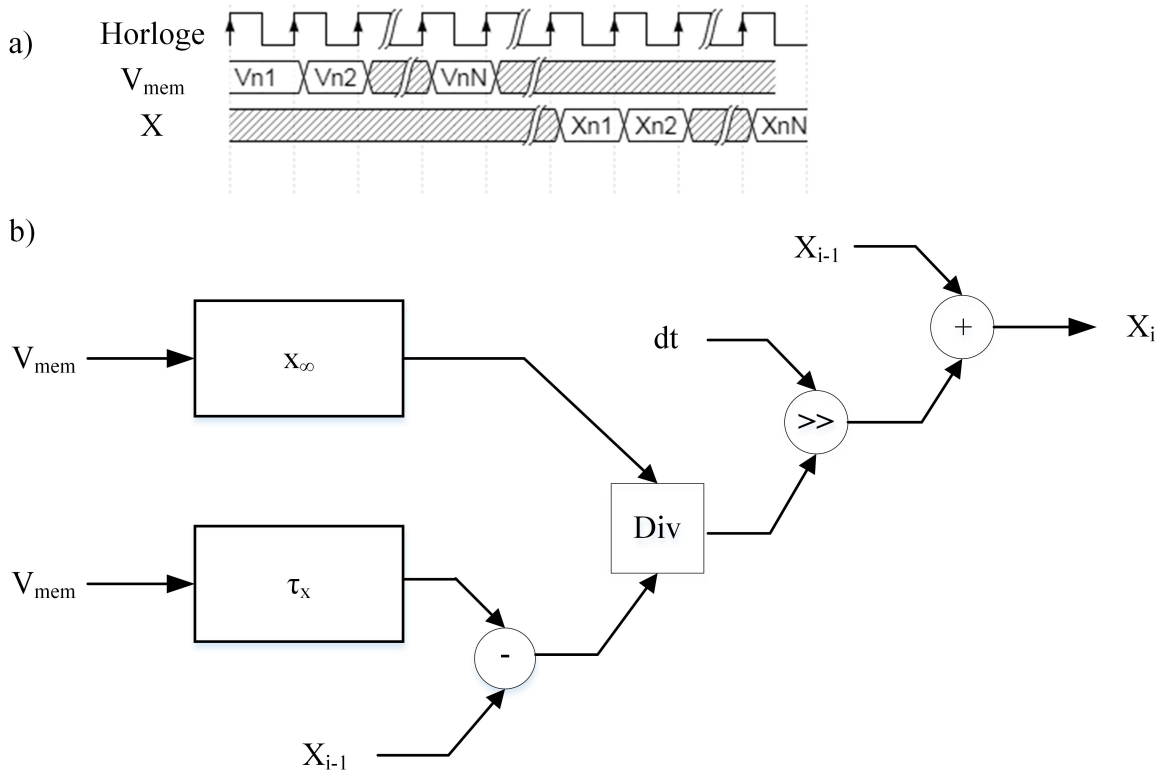

FIGURE 4.17 – Propagation de différentes valeurs de tensions de membranes (V_{mem}) correspondant à plusieurs neurones ; A) le chronogramme représente la propagation de la tension de membrane correspondant au neurone n1 jusqu'au neurone nN avec N le nombre de neurone maximum. X représente la valeur d'une variable d'activation ou d'inactivation ; B) Représentation de l'arbre de calcul sous forme de pipeline du calcul d'une variable d'activation ou d'inactivation.

Pour finir sur les performances en termes de délais et de temps de propagation dans les systèmes numériques, il faudra toujours favoriser les composants dédiés du FPGA. Ils

seront plus efficaces, optimaux et consommeront moins d'énergie que les algorithmes et prendront moins de ressources sur le FPGA.

3. Modèle proie-prédateur

Les fonctions biologiques sont issues de plusieurs substances chimiques liées dans des réseaux biochimiques complexes. De plus, de nombreux réseaux biochimiques affichent comme dynamique une oscillation [ELOW 2000]. Il est important de représenter la dynamique d'un réseau depuis sa cinétique jusqu'à sa topologie. Cependant, la simulation de ce type de réseau devient complexe à mesure que la taille des paramètres augmente ce qui a pour effet d'induire de long temps de calculs. Les architectures CPU conventionnelles ne sont pas adaptées pour faire du calcul massivement parallèle contrairement au FPGA. Ce dernier est une solution présentant les mêmes avantages que dans le cadre des calculs de réseaux de neurones. Le fonctionnement du système est représenté sur la figure 4.18 issue de [THAN 2019]. Elle montre en A l'architecture du réseau biochimique, en B la réaction de polymérisation-dépolymérisation de l'ADN constituant le réseau et en C et D la dynamique du système.

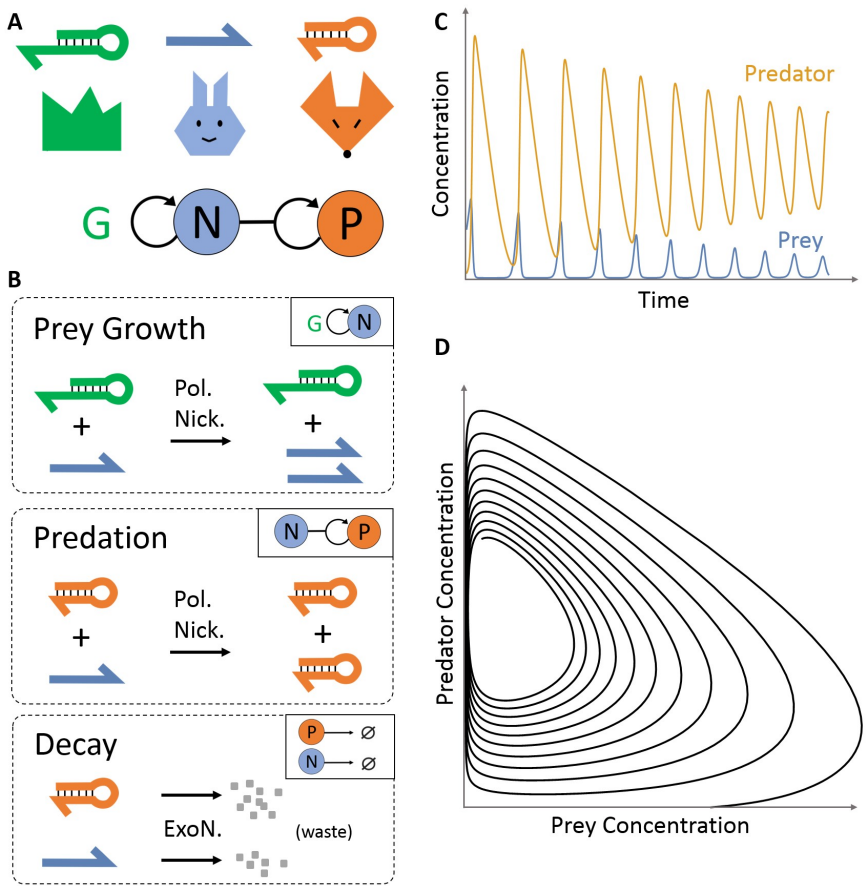

FIGURE 4.18 – Figure récupérée de [THAN 2019] représentant le fonctionnement du système. A) Architecture du réseau biochimique du système proie-prédateur; B) Réaction de polymérisation-dépolymérisation de l'ADN constituant le réseau; C) Comportement dynamique du système; D) Trajectoire 2D dans l'espace de concentration

Cette relation est représentée mathématiquement par les équations modifiées de Lotka-Volterra [FUJI 2013] et représentées par les équations différentielles 4.4 et 4.5.

$$\frac{dn}{dt} = \frac{gn}{1+\beta gn} - pn - \frac{\lambda\delta n}{1+p} \tag{4.4}$$

$$\frac{dp}{dt} = pn - \frac{\delta p}{1+p} \tag{4.5}$$

avec le premier terme de l'équation 4.4 modélise l'autoréplication de la proie n, le deuxième terme modélise la consommation de la proie n par le prédateur p, et, le dernier terme modélise la dégradation de la proie n par l'exonucléase comme pour l'équation 4.5. β, λ, δ et g sont des paramètres biochimiques du système proie-predateur.

4. Optimisation combinatoire

[BLUM 2003] décrit la résolution de problèmes d'optimisation comme étant la recherche des meilleures configurations d'un jeu de variables pour atteindre certains buts. Il décrit également les problèmes d'optimisation combinatoire comme étant une sous-classe des problèmes d'optimisation dont les variables sont encodées sous forme de variables discrètes. Les applications concernent la finance [LAUGH 1970], la sélection de projets [RHYS 1970], les analyses de cluster [RAO 1971], les analyses économiques [HAMM 1971], la gestion du trafic routier [WITZ 1975] et la conception assistée par ordinateur [KRAR 1978].

4.1 Max-cut

[KOCH 2014] review les différents types de problème et les différentes applications touchant aux problèmes d'optimisation combinatoire ainsi que les méthodes existantes permettant de les résoudre. [LELEU 2019] propose une alternative aux à la version mean-field de la méthode de recuit simulé [BILB 1989] qui est basée sur une dynamique similaire aux réseaux de neurones. Cette méthode offre une meilleure convergence aux solutions optimales (équations 4.6 à 4.8).

$$\frac{dx_i}{dt} = (-1+p)x_i - x_i^3 + \epsilon e_i \sum_{j \neq i} w_{ij} x_j \tag{4.6}$$

$$\frac{de_i}{dt} = -\beta(t)[x_i^2 - a]e_i \tag{4.7}$$

$$\frac{d\beta}{dt} = \lambda \tag{4.8}$$

avec w_{ij} une matrice de taille variable et x_j un vecteur de taille variable tel que le nombre d'élément du vecteur est égal au nombre d'éléments d'une ligne de la matrice w_{ij}, p et a sont représentés par les équations 4.9 à 4.10 et λ est un paramètre de l'équation différentielle 4.8.

$$a(\Delta H) = \alpha + \rho \tanh(\Delta H) \tag{4.9}$$

$$p(\Delta H) = \pi - \rho \tanh(\Delta H) \tag{4.10}$$

avec H l'Hamiltonien d'Ising représenté par l'équation 4.11, α, π et ρ sont des paramètres constants.

$$H = -\frac{1}{2} \sum_i \sum_j w_{ij} \sigma_i \sigma_j \tag{4.11}$$

ou, w_{ij} est la matrice décrite précédemment et σ, le signe du vecteur x.

Ces équations ressemblent fortement aux équations de calcul de réseaux de neurones. La matrice w_{ij} représente la connectivité comme les neurones peuvent l'être.

4.2 Description du système

Architecture générale

L'architecture générale du système est représentée par la figure 4.19 et est constituée d'une PLL pour adapter la fréquence d'horloge au système. Le système est aussi constitué de deux modules pour la transmission et l'émission d'informations par UART. Les paramètres constants tels que ceux présentés dans les équations précédentes, les éléments du vecteur x et de la matrice w, le pas de calcul et le nombre de pas à effectuer sont des paramètres transférés par UART puis sauvegardé en mémoire (RAM).

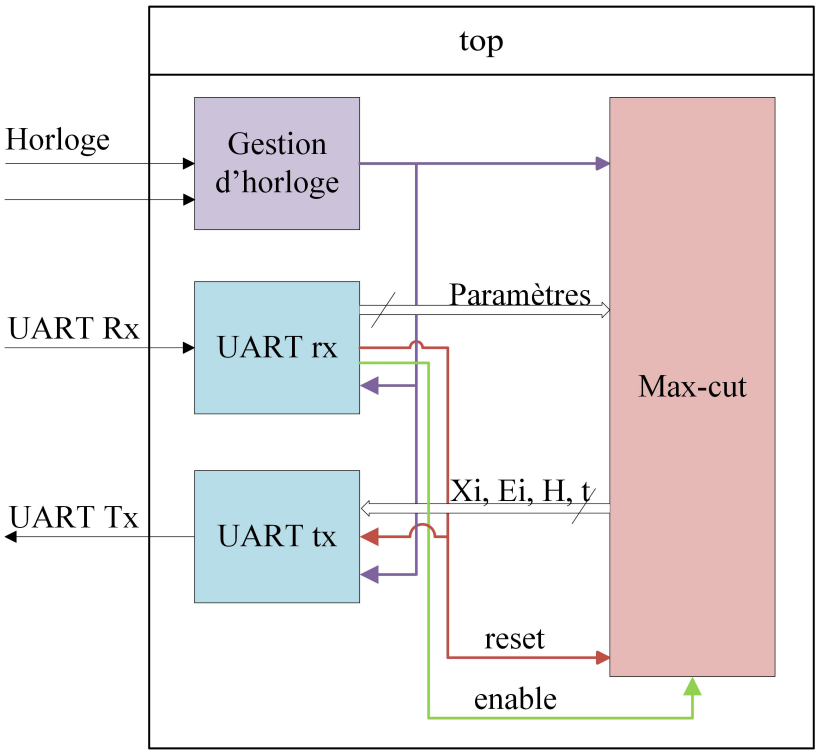

FIGURE 4.19 – Architecture générale du système avec une PLL, deux modules servant respectivement à émettre et transmettre des informations via UART et le module principal de calcul

Le codage des données a été fixé à 5 bits pour la partie entière et 18 bits pour la partie décimale pour un total de 24 bits en incluant le bit de signe. Ce choix a été fait car les simulations ont montré que le système tendait vers l'instabilité numérique lorsque la partie décimale des vecteurs passent en dessous de 14 bits.

Machine d'état principale du calcul du Max-cut

Le cœur de calcul principal est, quant à lui, constitué d'une machine d'état principale dont l'état IDLE est l'état qui par défaut reçoit les paramètres UART. La machine d'état est

découpée en quatre parties. Il y a tout d'abord la partie IDLE qui attend la réception d'un signal d'activation généré par le module UART. Ensuite, la partie calcul qui effectue en parallèle un produit matrice vecteur, les calculs de x et de e (Figure 4.20), le calcul d'Ising et les calculs de a et de p (Figure 16) qui sont tous entièrement créés sous forme de pipeline pour permettre une vitesse de calcul optimale. Lorsque le calcul du produit de la matrice w_{ij} par le vecteur x_i est terminé, un signal est envoyé à la machine d'état principale pour qu'elle continue vers l'étape suivante. Celle-ci sert à sauvegarder et rassembler les résultats des calculs. Une fois que les données ont été sauvegardées, la machine d'état passe dans un mode d'envoi de données via UART.

Produits matrice vecteur

Le module utilise la méthode décrite dans le chapitre 3. Dans un premier temps le vecteur est multiplié par chaque ligne de la matrice en utilisant la méthode citée et, dans un second temps, chaque terme de chaque ligne de la matrice sont additionnés en utilisant les DSP en mode additionneurs et cascades. Les DSP48E2 présents dans les FPGA ultrascale sont placés deux par deux pour faciliter le mode cascade. Il est donc possible d'avoir un DSP qui additionne 3 valeurs et un autre DSP qui additionne 2 autres valeurs avec le résultat du DSP précédent. L'intérêt d'utiliser cette méthode est d'éviter d'ajouter du routage au circuit ce qui a pour effet d'augmenter les performances du système.

Calcul de x et e

Les calculs de x et de e (équations 4.6 et 4.7) sont relativement classiques. Ils sont organisés en pipeline et partagent le calcul x^2 afin d'éviter les répétitions et les pertes de ressources et d'énergie. Le calcul s'effectue en 9 cycles d'horloge et se lance en même temps que le signal d'activation de la machine d'état présentée dans la partie précédente. Le module est entièrement configuré pour n'utiliser que des DSP pour une meilleure optimisation. La seule particularité est la mise en place d'une détection d'overflow à la fin du calcul de E car les valeurs de ce dernier peuvent dépasser le domaine de définition admissible par les vecteurs de données.

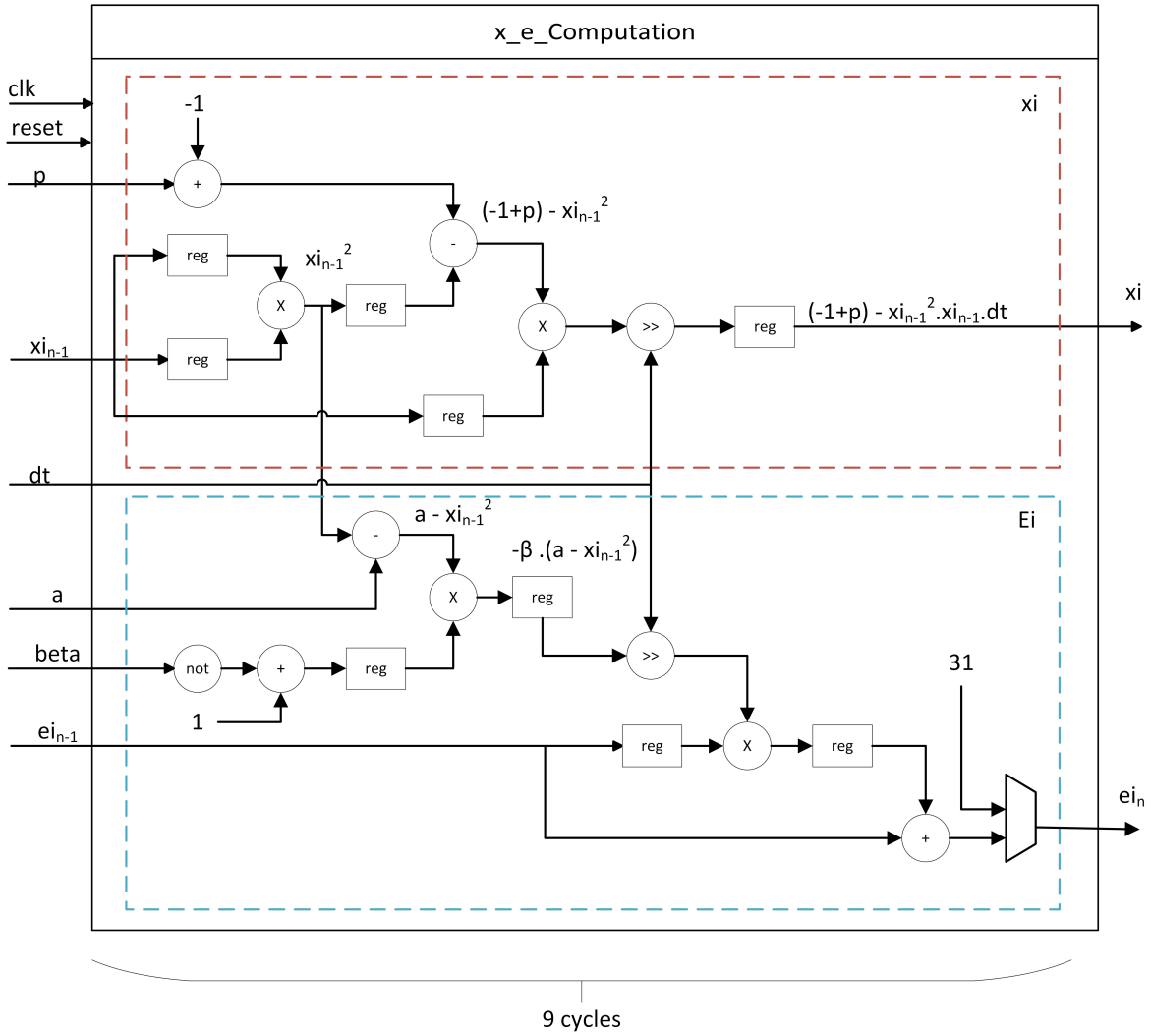

FIGURE 4.20 – Architecture de l'implémentation des calculs de e et x. Chaque multiplication représente un DSP configuré de sorte à avoir un pipeline en entrée et en sortie ce qui a pour effet d'allonger la durée du calcul de 2 cylces supplémentaires par multiplications. Les premiers résultats arrivent 9 cycles après le signal d'activation de la machine d'état.

Calcul des termes a et p

Les calculs de a et p sont aussi entièrement pipeline et utilisent exclusivement les DSP. Les calculs de tangente hyperbolique sont réalisés à partir de l'interpolation de Lagrange présenté dans le chapitre précédent. L'erreur induite par l'interpolation est acceptable dans le cadre des calculs de a et p (équations 4.9 et 4.10) ce qui est démontré par les résultats présentés sur la figure 4.13 de la partie suivante.

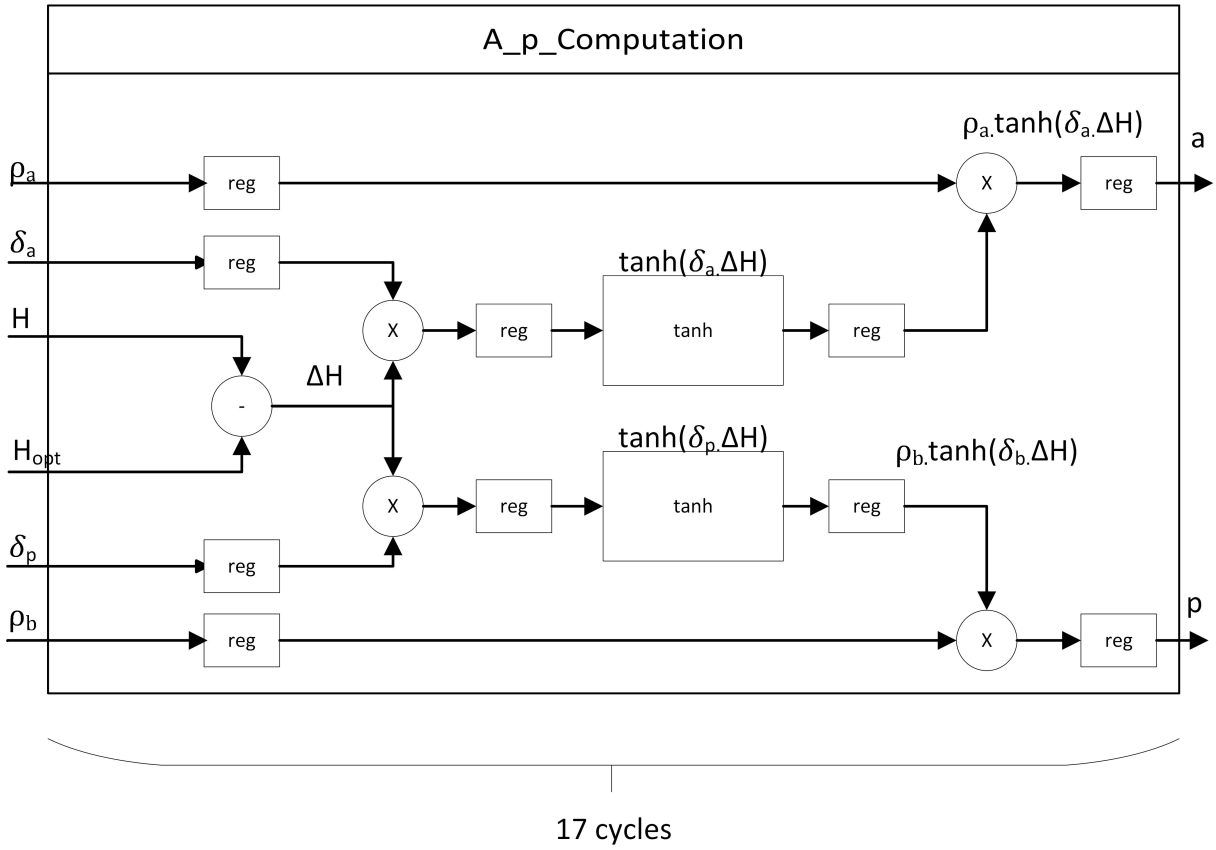

FIGURE 4.21 – Architecture des calculs de a et p en utilisant l'interpolation de Lagrange pour effectuer la tangente hyperbolique.