



HAL
open science

Neurone abstrait : une formalisation de l'intégration dendritique et ses propriétés algébriques

Ophélie Guinaudeau

► **To cite this version:**

Ophélie Guinaudeau. Neurone abstrait : une formalisation de l'intégration dendritique et ses propriétés algébriques. Bio-informatique [q-bio.QM]. COMUE Université Côte d'Azur (2015 - 2019), 2019. Français. NNT : 2019AZUR4001 . tel-02389736

HAL Id: tel-02389736

<https://theses.hal.science/tel-02389736v1>

Submitted on 2 Dec 2019

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 DE DOCTORAT

Neurone abstrait :
une formalisation de l'intégration dendritique
et ses propriétés algébriques

Ophélie GUINAUDEAU

Laboratoire d'Informatique, Signaux et Systèmes de Sophia Antipolis (I3S)
UMR7271 UNS CNRS

**Présentée en vue de l'obtention du
grade de docteur en Informatique
d'Université Côte d'Azur**

Dirigée par :
Gilles Bernot

Co-encadrée par :
Franck Grammont et Alexandre Muzy

Soutenue le : 11 janvier 2019

Devant le jury, composé de :

Gilles Bernot, PR, Université Côte d'Azur
Daniel Gaffé, MCF, Université Côte d'Azur
Franck Grammont, MCF, Université Côte d'Azur
François Képès, DR, Synovance
Pascale Le Gall, PR, École CentraleSupélec
Timothée Masquelier, CR, Université de Toulouse
Christian Michel, PR, Université de Strasbourg
Alexandre Muzy, CR, Université Côte d'Azur

Neurone abstrait : une formalisation de l'intégration dendritique et ses propriétés algébriques

Composition du jury :

Rapporteurs

Pascale Le Gall, Professeure des Universités, École CentraleSupélec
Christian Michel, Professeur des Universités, Université de Strasbourg

Examineurs

Gilles Bernot, Professeur des Universités, Université Côte d'Azur
Franck Grammont, Maître de conférences, Université Côte d'Azur
François Képès, Directeur de Recherche, Synovance
Timothée Masquelier, Chargé de recherche CNRS, Université de Toulouse
Alexandre Muzy, Chargé de recherche CNRS, Université Côte d'Azur

Invité

Daniel Gaffé, Maître de conférences, Université Côte d'Azur

Remerciements

Je tiens tout d'abord à remercier les membres de mon jury pour leur participation à ma soutenance de thèse. En particulier, je remercie Pascale Le Gall et Christian Michel d'avoir pris le temps de rapporter sur ma thèse.

Mes plus grands remerciements vont à mon directeur de thèse, Gilles, avec qui j'ai eu la chance de travailler pendant ces trois années. Gilles m'a appris énormément d'un point de vue scientifique, parfois dans des domaines que je n'aurais jamais pensé explorer. Et plus que des connaissances techniques, il m'a inculqué une véritable démarche scientifique, une manière de penser. Plus personnellement, je le remercie grandement pour ses conseils, ses encouragements, et pour la confiance qu'il a parfois dû avoir pour 2!

Je remercie énormément Franck pour avoir initié ce projet de recherche et Alexandre pour m'avoir fait confiance. Merci aussi à Daniel pour sa gentillesse et pour avoir donné de nouvelles perspectives à mon travail. J'adresse également mes remerciements aux membres du laboratoire I3S et de l'école Doctorale grâce à qui j'ai fait ma thèse dans un cadre privilégié. Merci de m'avoir fait confiance en m'accordant l'allocation et de m'avoir offert la possibilité de m'impliquer dans la vie scientifique.

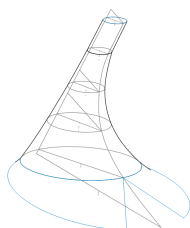
Je tiens à remercier certaines personnes, rencontrées durant ma scolarité, et sans qui je n'en serais pas là aujourd'hui. Je remercie particulièrement Frédéric Precioso qui m'a donné le goût de l'informatique, m'a soutenue dans mon projet de thèse et m'a gentiment orientée vers Alexandre. Merci aussi à Carole Rovere pour m'avoir permis de travailler sur mon premier projet de bio-informatique et pour avoir accepté de faire partie de mon comité de suivi de thèse quelques années plus tard.

Un grand merci à tous ceux grâce à qui j'ai fait ma thèse en bonne compagnie. Merci aux doctorants : merci à Émilien et Ben qui ont su me guider et me conseiller dès mon arrivée, merci à John avec qui j'ai passé de très bons moments (surtout en formations), et merci à Cyrille pour les discussions très intéressantes que je regrette de ne pas avoir eu plus souvent. Merci aussi à Maxime, Guillaume, Heytem, Steven, Gaël et Manu, Assia, Eman et Oussama. Merci également aux deux petits derniers, Nico et Lætitia, dont j'ai vraiment apprécié la présence quand le labo était un peu vide pendant la rédaction! Merci également à Rémy et Lyès pour leur enthousiasme, leur implication, et pour avoir pris la suite de l'ADSTIC. Et puis, merci aux nombreux stagiaires qui ont apporté de la nouveauté et de la bonne humeur! Plus largement, merci à toute l'(ex-)équipe MDSC et surtout à Jean-Paul pour sa sympathie et ses bons conseils. Merci aussi à Cinzia, Elisabetta, Joëlle, Patrick, Lauriane, Arnaud, Jean-Charles et Marie pour sa bonne volonté et sa motivation. Je remercie aussi Hélène pour sa bienveillance et son implication dans l'enseignement, et Magali qui a toujours été disponible, efficace et souriante.

Je remercie énormément mes amis : en particulier Jules, Pauline, Anaïs et mes copines de thèse, Estelle et Ingrid, pour leur soutien, leur compréhension et pour avoir subi ma période de rédaction.

Mes derniers remerciements vont à ma famille, ma sœur et mes parents qui ont toujours cru en moi et ont toujours été prêts à m'aider quand ils le pouvaient (*cf. Figure ci-dessous*). Enfin, un très grand merci à Matthieu qui m'a soutenue pendant cette *interminable* thèse.

Si je devais remercier toutes les personnes grâce à qui j'ai pu mener à bien cette thèse, je devrais probablement rajouter un chapitre. Je vais donc terminer ces remerciements avec un grand merci à toutes les personnes que je n'ai pas citées ici, et qui ont contribué de près ou de loin à cet accomplissement.



Perspective.

Titre : Neurone abstrait : une formalisation de l'intégration dendritique et ses propriétés algébriques.

Résumé : Les neurones biologiques communiquent par le biais d'impulsions électriques, appelées *spikes*, et les fonctions cérébrales émergent notamment de la coordination entre les réceptions et émissions de ces spikes. Par ailleurs, il est largement admis que la fonction de chaque neurone dépend de sa morphologie. Les *dendrites* conditionnent l'intégration spatio-temporelle des spikes reçus et influent sur les temps d'occurrence des spikes émis. Elles sont donc fondamentales pour l'étude *in silico* des mécanismes de coordination, et en particulier pour l'étude des *assemblées de neurones*. Les modèles de neurones existants prenant en compte les dendrites, sont généralement des modèles mathématiques détaillés, souvent à base d'équations différentielles, dont la simulation nécessite des ressources de calculs importantes. De plus, leur complexité intrinsèque rend difficile l'analyse et les preuves sur ces modèles. Dans cette thèse, nous proposons un modèle de neurone intégrant des dendrites d'une manière *abstraite*. Dans l'objectif d'ouvrir la porte aux *méthodes formelles*, nous établissons une définition rigoureuse du cadre de modélisation et mettons en évidence des *propriétés algébriques* remarquables de l'intégration dendritique. Nous avons notamment démontré qu'il est possible de réduire la structure d'un neurone en préservant sa fonction d'entrée/sortie. Nous avons ainsi révélé des *classes d'équivalence* dont nous savons déterminer un représentant canonique. En s'appuyant sur la théorie des catégories et par des *morphismes de neurones* judicieusement définis, nous avons ensuite analysé plus finement ces classes d'équivalence. Un résultat surprenant découle de ces propriétés : un simple ajout de *délais* dans les modèles informatiques de neurones permet de prendre en compte une intégration dendritique abstraite, sans représenter explicitement la structure arborescente des dendrites. À la racine de l'arborescence dendritique, la modélisation du soma contient inévitablement une équation différentielle lorsque l'on souhaite préserver l'essence du fonctionnement biologique. Ceci impose de combiner une *vision analytique* avec la *vision algébrique*. Néanmoins, grâce à une étape préalable de discrétisation temporelle, nous avons également implémenté un neurone complet en *Lustre* qui est un langage formel autorisant des preuves par model checking. Globalement, nous apportons dans cette thèse un premier pas encourageant vers une formalisation complète des neurones, avec des propriétés remarquables sur l'intégration dendritique.

Mots clés : modélisation formelle, neurone, dendrites, propriétés algébriques, délais, Lustre

Title : Abstract neuron : formalizing dendritic integration and algebraic properties.

Abstract : Biological neurons communicate by means of electrical impulses, called *spikes*. Brain functions emerge notably from reception and emission coordination between those spikes. Furthermore, it is widely accepted that the function of each neuron depends on its morphology. In particular, *dendrites* perform the spatio-temporal integration of received spikes and affect the occurrence of emitted spikes. Dendrites are therefore fundamental for *in silico* studies of coordination mechanisms, and especially for the study of so-called *neuron assemblies*. Most of existing neuron models taking into account dendrites are detailed mathematical models, usually based on differential equations, whose simulations require significant computing resources. Moreover, their intrinsic complexity makes difficult the analysis and proofs on such models. In this thesis, we propose an *abstract* neuron model integrating dendrites. In order to pave the way to *formal methods*, we establish a rigorous definition of the modeling framework and highlight remarkable *algebraic properties* of dendritic integration. In particular, we have demonstrated that it is possible to reduce a neuron structure while preserving its input/output function. We have thus revealed *equivalence classes* with a canonical representative. Based on category theory and thanks to properly defined *neuron morphisms*, we then analyzed these equivalence classes in more details. A surprising result derives from these properties : simply adding *delays* in neuron computational models is sufficient to represent an abstract dendritic integration, without explicit tree structure representation of dendrites. At the root of the dendritic tree, soma modeling inevitably contains a differential equation in order to preserve the biological functioning essence. This requires combining an *analytical vision* with the *algebraic vision*. Nevertheless, thanks to a preliminary step of temporal discretization, we have also implemented a complete neuron in *Lustre* which is a formal language allowing proofs by model checking. All in all, we bring in this thesis an encouraging first step towards a complete neuron formalization, with remarkable properties on dendritic integration.

Keywords : formal modelling, neuron, dendrites, algebraic properties, delays, Lustre

TABLE DES MATIÈRES

| | |
|--|-----------|
| Introduction | 9 |
| I Contexte pluridisciplinaire | 19 |
| 1 Prérequis biologiques | 21 |
| 1.1 Le neurone : unité élémentaire du système nerveux | 21 |
| 1.1.1 Une structure adaptée à sa fonction | 21 |
| 1.1.2 La perméabilité membranaire à l'origine de l'activité électrique | 23 |
| 1.1.3 La transmission de l'influx nerveux | 26 |
| 1.1.4 Les dendrites et leurs propriétés | 30 |
| 1.2 Quelques mots sur la plasticité | 38 |
| 1.3 Théories synchrones et asynchrones du fonctionnement cérébral | 40 |
| 1.3.1 Les assemblées de neurones | 41 |
| 1.3.2 Les groupes de neurones polychrones | 43 |
| 1.3.3 Le codage temporel : point commun des deux théories | 45 |
| 2 Prérequis mathématiques et informatiques | 47 |
| 2.1 Fonctions mathématiques | 47 |
| 2.1.1 Concepts élémentaires utiles | 47 |
| 2.1.2 Les équations différentielles | 50 |
| 2.1.3 Quelques principes d'induction utiles | 54 |
| 2.1.4 Les segments | 55 |
| 2.2 Algèbre générale | 56 |
| 2.2.1 Quelques rappels d'algèbre classique | 56 |
| 2.2.2 Structures algébriques | 57 |
| 2.3 Théorie des catégories | 59 |
| 2.3.1 Catégories et diagrammes | 60 |

TABLE DES MATIÈRES

| | | |
|-----------|--|------------|
| 2.3.2 | Propriétés des morphismes | 61 |
| 2.3.3 | Propriétés des objets | 62 |
| 2.3.4 | Propriétés des catégories | 63 |
| 2.3.5 | Foncteurs et transformations naturelles | 64 |
| 2.4 | Langages synchrones et model checking | 65 |
| 2.4.1 | Les langages synchrones | 65 |
| 2.4.2 | Le langage Lustre | 66 |
| 2.4.3 | Le model checking pour Lustre | 69 |
| 3 | État de l'art et motivations | 73 |
| 3.1 | Modèles de neurones uniques | 74 |
| 3.1.1 | Modèles bio-inspirés versus modèles biologiquement plausibles | 74 |
| 3.1.2 | Modèles fréquentiels versus modèles impulsionsnels | 80 |
| 3.1.3 | Modèles ponctuels versus modèles morphologiques | 83 |
| 3.1.4 | Modèles déterministes versus modèles probabilistes | 85 |
| 3.2 | Méthodes formelles pour les systèmes biologiques | 86 |
| 3.2.1 | Le formalisme discret de René Thomas pour les réseaux génétiques | 86 |
| 3.2.2 | Les réseaux de Petri pour les réseaux métaboliques | 89 |
| 3.2.3 | Les formalismes à base de règles | 91 |
| 3.2.4 | Méthodes formelles pour les neurosciences | 92 |
| 3.3 | Nos motivations | 95 |
| II | Neurones abstraits avec dendrites : notre cadre de modélisation | 99 |
| 4 | Définition du cadre de modélisation | 101 |
| 4.1 | Description statique d'un neurone | 103 |
| 4.1.1 | Les synapses | 104 |
| 4.1.2 | Les compartiments | 106 |
| 4.1.3 | Le soma | 106 |
| 4.1.4 | Le neurone complet | 109 |
| 4.1.5 | État d'un neurone | 112 |
| 4.2 | Description de la dynamique d'un neurone | 115 |
| 4.2.1 | Signaux discrets | 115 |
| 4.2.2 | Du discret au continu : la dynamique des synapses | 116 |
| 4.2.3 | Des synapses au soma : la dynamique dendritique | 120 |
| 4.2.4 | Intégration des signaux : la dynamique du soma | 126 |
| 4.2.5 | Dynamique globale du neurone et signal de sortie discret | 129 |
| 5 | Résultats fondamentaux | 131 |
| 5.1 | Les neurones réduits et leurs propriétés | 131 |
| 5.1.1 | Fonction de réduction | 132 |
| 5.1.2 | Reconstruction d'état | 133 |

| | | |
|---|--|----------------|
| 5.1.3 | Théorème de réduction | 135 |
| 5.2 | Les neurones normalisés : propriétés et limitation | 141 |
| 5.2.1 | Fonction de normalisation | 142 |
| 5.2.2 | Dé-normalisation d'état | 143 |
| 5.2.3 | Théorème de normalisation | 145 |
| 5.2.4 | Limitation | 150 |
| 5.3 | Portée de nos résultats | 151 |
| 5.3.1 | Interprétation biologique | 152 |
| 5.3.2 | Apport à l'informatique bio-inspirée | 153 |
| 6 | Une vision purement algébrique | 155 |
| 6.1 | La catégorie des neurones | 156 |
| 6.1.1 | Définitions préliminaires | 156 |
| 6.1.2 | Morphismes de neurones | 157 |
| 6.1.3 | Propriétés des morphismes | 161 |
| 6.1.4 | Propriétés des objets | 162 |
| 6.2 | Les catégories de signaux d'entrée | 165 |
| 6.2.1 | Signaux d'entrée discrets | 165 |
| 6.2.2 | Signaux d'entrée continus | 168 |
| 6.3 | La catégorie des états dendritiques | 170 |
| 6.4 | Préservation de la fonction d'entrée/sortie des neurones par morphisme | 175 |
| III Implémentation discrète du cadre de modélisation et extension vers la plasticité | | 179 |
| 7 | Discrétisation temporelle du modèle | 181 |
| 7.1 | Choix du pas de temps | 182 |
| 7.2 | Discrétisation aux synapses | 182 |
| 7.3 | Discrétisation aux compartiments | 183 |
| 7.4 | Discrétisation au soma | 184 |
| 8 | Prototypage avec Lustre | 185 |
| 8.1 | Encodage du modèle | 185 |
| 8.1.1 | Encodage d'une synapse | 185 |
| 8.1.2 | Encodage d'un compartiment | 187 |
| 8.1.3 | Encodage d'un soma | 188 |
| 8.1.4 | Encodage d'un neurone | 191 |
| 8.1.5 | Encodage d'un réseau | 192 |
| 8.2 | Encodage et vérification de propriétés par model checking | 192 |
| 8.2.1 | Preuve de concept | 192 |
| 8.2.2 | Limitations | 195 |
| 8.3 | Exemple de simulation du modèle Lustre | 196 |

TABLE DES MATIÈRES

| | | |
|----------|--|------------|
| 9 | Introduction de la plasticité synaptique | 199 |
| 9.1 | Vue d'ensemble | 199 |
| 9.2 | Formalisation | 201 |
| 9.3 | Discussion | 205 |
| | Conclusion et perspectives | 207 |
| | Annexes | 213 |
| A | Code Lustre - partie 1 | 216 |
| B | Code Lustre - partie 2 | 221 |
| C | Code Lustre - partie 3 | 223 |
| D | Code Java permettant de générer du code Lustre | 224 |
| E | Complément Figure 8.8, Chapitre 8 | 226 |
| | Bibliographie | 226 |

LISTE DES FIGURES

| | | |
|------|---|----|
| 1 | Dessin d'un neurone de Purkinje du cortex cérébelleux du chat | 11 |
| 1.1 | Structure schématique d'un neurone archétypique | 21 |
| 1.2 | Variabilité structurelle des neurones | 22 |
| 1.3 | Enregistrement électrophysiologique d'un neurone cortical pyramidal | 23 |
| 1.4 | Potentiel de repos | 24 |
| 1.5 | Potentiel d'action et courants ioniques associés | 25 |
| 1.6 | Propagation du potentiel d'action dans l'axone | 26 |
| 1.7 | Représentation schématique d'une synapse excitatrice | 28 |
| 1.8 | Sommation spatio-temporelle | 29 |
| | (a) Sommation temporelle | 29 |
| | (b) Sommation spatiale | 29 |
| 1.9 | Propagation décroissante du potentiel post-synaptique dans les dendrites . | 30 |
| 1.10 | Atténuation des potentiels post-synaptiques | 33 |
| | (a) Extrait de [Stuart and Spruston, 1998] | 33 |
| | (b) Extrait de [Spruston, 2008] | 33 |
| 1.11 | Potentiels d'action somatique et dendritique | 35 |
| 1.12 | Neurones pyramidaux du néocortex, chez différentes espèces de vertébrés . . | 36 |
| 1.13 | Neurone pyramidal du CA1 | 37 |
| 1.14 | Fenêtre temporelle pour l'induction de LTP et LTD | 40 |
| 1.15 | Représentation schématique du problème de liage | 42 |
| 1.16 | Représentation schématique d'un groupe de neurones polychrones | 44 |
| 2.1 | Injection | 47 |
| 2.2 | Surjection | 48 |
| 2.3 | Bijection | 48 |
| 2.4 | Exemple de fonction uniformément continue | 49 |
| 2.5 | Exemple de fonction dérivable à droite | 49 |

LISTE DES FIGURES

| | | |
|------|--|-----|
| 2.6 | Exemple de fonction k-lipschitzienne | 50 |
| 2.7 | Support visuel pour les axiomes des catégories | 61 |
| | (a) Axiome <i>C0</i> | 61 |
| | (b) Axiome <i>C1</i> | 61 |
| | (c) Axiome <i>C2</i> | 61 |
| 2.8 | Exemple de diagramme commutatif | 61 |
| 2.9 | Monomorphisme | 62 |
| 2.10 | Épimorphisme | 62 |
| 2.11 | Isomorphisme | 62 |
| 2.12 | Objet initial et objet terminal d'une catégorie | 63 |
| 2.13 | Foncteur | 64 |
| 2.14 | Transformation naturelle | 65 |
| 2.15 | Schéma logique d'un multiplexeur | 67 |
| 2.16 | Fonctionnement de Luciole | 69 |
| | (a) Fenêtre de simulation avec <i>Luciole</i> | 69 |
| | (b) Chronogramme avec l'outil <i>Sim2chro</i> | 69 |
| 3.1 | Représentation de la membrane neuronale par un circuit électrique | 75 |
| 3.2 | Reproduction des données expérimentales par le modèle Hodgkin-Huxley | 76 |
| 3.3 | Représentation schématique du neurone de McCulloch et Pitts | 78 |
| 3.4 | Calcul des fonctions logiques <i>ET</i> et <i>OU</i> par un neurone formel | 79 |
| 3.5 | Représentation de la membrane neuronale du Leaky Integrate-and-Fire | 81 |
| 3.6 | Exemple simple d'un graphe d'interaction et de son graphe d'états associés | 87 |
| 3.7 | Exemple de réseau de Petri | 89 |
| 3.8 | Métabolisme du fructose et sa représentation en réseau de Petri | 90 |
| | (a) Métabolisme du fructose dans le foie. | 90 |
| | (b) Représentation par réseau de Petri. | 90 |
| 3.9 | Les archétypes | 93 |
| 3.10 | Automate temporisé pour modéliser un Leaky Integrate and Fire | 95 |
| 4.1 | Membrane représentée comme un circuit électrique simplifié | 102 |
| 4.2 | Représentation schématique de notre modèle de neurone | 103 |
| 4.3 | Exemples de fonctions synaptiques | 105 |
| | (a) Cas positif affine par morceaux. | 105 |
| | (b) Cas négatif affine par morceaux. | 105 |
| | (c) Cas positif à double exponentielle. | 105 |
| | (d) Cas négatif à double exponentielle. | 105 |
| 4.4 | Variabilité des potentiels post-synaptiques | 105 |
| 4.5 | Évolution du seuil en fonction du temps écoulé depuis le dernier spike | 107 |
| 4.6 | Exemples de fonctions de borne | 108 |
| | (a) Fonction de borne affine par morceaux | 108 |
| | (b) Fonction de borne exponentielle | 108 |

| | | |
|------|---|-----|
| 4.7 | Représentation schématique de notre modèle de neurone | 109 |
| 4.8 | Prédécesseurs d'un compartiment dans un neurone | 110 |
| 4.9 | Définition alternative d'un neurone | 111 |
| (a) | Notre définition | 111 |
| (b) | Définition alternative | 111 |
| 4.10 | Définition alternative d'un neurone : cas particulier | 111 |
| (a) | Notre définition | 111 |
| (b) | Définition alternative (cas particulier) | 111 |
| 4.11 | Zoom sur une « portion » de neurone | 112 |
| 4.12 | Abstraction du temps et de l'espace | 113 |
| 4.13 | Trace d'un signal élémentaire sur une synapse | 117 |
| 4.14 | Réponses à des expériences de patch clamp en courant imposé | 118 |
| 4.15 | Comportement exponentiel de la somme des traces de spikes | 118 |
| 4.16 | Exemples de signaux ayant des traces différentes mais égales en zéro | 120 |
| (a) | Exemple croissant. | 120 |
| (b) | Exemple décroissant. | 120 |
| 4.17 | Exemple d'un shift de durée Δ_c (premier cas) | 121 |
| (a) | État d'une « portion » de neurone au temps t | 121 |
| (b) | État d'une « portion » de neurone au temps $t + \Delta_c$ | 121 |
| 4.18 | Exemple d'un shift d'une durée Δ_c (second cas) | 122 |
| (a) | État d'une « portion » de neurone au temps t | 122 |
| (b) | État d'une « portion » de neurone au temps $t + \Delta_c$ | 122 |
| 4.19 | Dynamique des dendrites par ε -shift | 123 |
| (a) | État d'une « portion » de neurone au temps t | 123 |
| (b) | État d'une « portion » de neurone au temps $t + \varepsilon$ | 123 |
| 4.20 | Illustration du Lemme 3 | 129 |
| 4.21 | Dynamique du soma d'un neurone | 130 |
| 5.1 | Exemples de neurones | 132 |
| (a) | Neurone non réduit | 132 |
| (b) | Neurone réduit | 132 |
| 5.2 | Fonction de réduction | 133 |
| (a) | Neurone non réduit N | 133 |
| (b) | Neurone réduit $\mathbb{P}(N)$ | 133 |
| 5.3 | Reconstruction de l'état des compartiments à partir de la forme réduite | 134 |
| 5.4 | Illustration du théorème de réduction | 136 |
| 5.5 | Exemple de fonction de borne normale | 142 |
| 5.6 | Exemple de normalisation | 143 |
| (a) | Neurone N | 143 |
| (b) | Neurone normalisé $P_{\mathbb{1}} = \mathbf{1}(N)$ | 143 |
| 5.7 | Illustration du théorème de normalisation | 145 |
| 5.8 | Illustration du corollaire du théorème de normalisation | 149 |

LISTE DES FIGURES

| | | |
|------|--|-----|
| 6.1 | Redistribution de compartiments | 159 |
| 6.2 | Factorisation de compartiments avec ajout (type 1) | 159 |
| 6.3 | Factorisation de compartiments sans ajout (type 2) | 160 |
| 6.4 | Renforcement des synapses | 160 |
| 6.5 | Renforcement du soma | 161 |
| 6.6 | Objet minimal de la catégorie des neurones | 162 |
| 6.7 | Illustration du corollaire 3 | 164 |
| 6.8 | Foncteur F_S | 166 |
| 6.9 | Transformation naturelle Δt -Shift | 168 |
| 6.10 | Transformation naturelle <i>Convert</i> | 170 |
| 6.11 | Illustration de la Définition 32 | 172 |
| 6.12 | Transformation naturelle <i>Init</i> | 174 |
| 6.13 | Illustration de la Proposition 5 | 175 |
| 6.14 | Diagramme commutatif | 176 |
| | | |
| 7.1 | Simulation de l'état d'une synapse excitatrice en réponse à un signal d'entrée | 183 |
| 7.2 | Discrétisation aux compartiments | 183 |
| | | |
| 8.1 | Calcul de l'état d'une synapse en fonction d'un signal d'entrée | 186 |
| 8.2 | Simulation du Nœud 1 par l'outil Luciole | 187 |
| 8.3 | Illustration du Nœud 2 | 187 |
| 8.4 | Simulation du Nœud 2 par l'outil Luciole | 188 |
| 8.5 | Simulation du Nœud 3 par l'outil Luciole | 190 |
| 8.6 | Exemple d'un neurone et de sa forme réduite | 193 |
| | (a) Neurone non réduit neuron1 | 193 |
| | (b) Neurone réduit neuron2 | 193 |
| 8.7 | Exemple de neurone réduit à 2 synapses identiques | 196 |
| 8.8 | Simulation du neurone illustré en Figure 8.7 avec l'outil Luciole | 197 |
| | | |
| 9.1 | Fonction basique de STDP | 200 |
| 9.2 | Fonction de d-STDP | 202 |
| 9.3 | Dynamique du soma d'un neurone | 203 |
| 9.4 | Dynamique enrichie d'un neurone | 205 |
| 9.5 | Évolution, sur une fenêtre temporelle choisie, de la valeur du potentiel p au soma (première ligne) et du signal de sortie O (dernière ligne) en réponse au signal d'entrée I (deuxième ligne) identiquement reçu sur chaque synapse. Les différentes valeurs de Δt correspondent au temps de traversée du compartiment variable de la Figure 8.7 du Chapitre 8. | 226 |

LISTE DES TABLES

| | | |
|-----|---|-----|
| 1.1 | Potentiel d'inversion des principaux ions | 24 |
| 2.1 | Propriétés fondamentales de l'addition et de la multiplication | 56 |
| 8.1 | Preuve d'équivalence de neurones par model checking (sans soma) | 194 |
| 8.2 | Preuve d'équivalence de neurones par model checking (avec soma) | 195 |

INTRODUCTION

Le cerveau est de loin l'organe le plus complexe du corps humain. Il occupe un rôle central dans son fonctionnement en réalisant de nombreuses fonctions, des plus primitives aux plus évoluées (contrôle du rythme cardiaque, de la respiration, du sommeil, des fonctions motrices, gestion des émotions, apprentissage, pensée, raisonnement, *etc.*).

Le *neurone*, objet principal de cette thèse, est l'unité structurelle et fonctionnelle de base du système nerveux. Le neurone permet la genèse et la communication de l'activité bioélectrique au sein de ce système. C'est une cellule qui possède un corps cellulaire (ou *soma*) et deux types de prolongements : des *dendrites* qui peuvent être très ramifiées, et un *axone* unique qui peut présenter des collatérales et une arborisation terminale (Figure 1). Les dendrites sont le lieu privilégié de la réception de l'*influx nerveux* tandis que l'axone est dédié à sa transmission aux autres neurones. Les points de connexion entre deux neurones sont appelés *synapses*. On distingue alors le neurone *pré-synaptique* qui transmet l'influx nerveux, du neurone *post-synaptique* qui le reçoit.

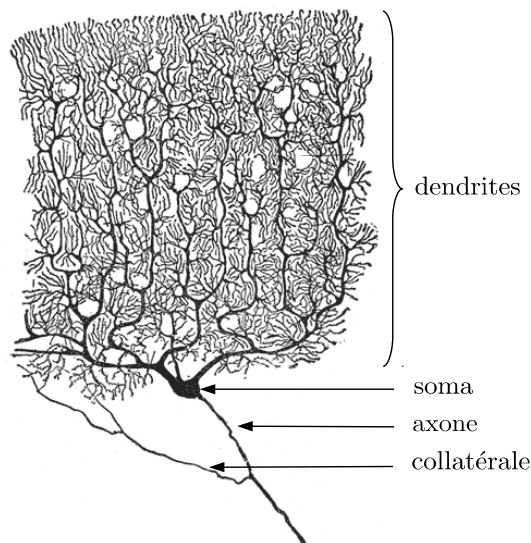


FIGURE 1 – Dessin d'un neurone de Purkinje du cortex cérébelleux du chat par Santiago Ramón y Cajal (fin du 19^{ième} siècle). Le neurone a été mis en évidence grâce à la coloration de Golgi qui permet de révéler des détails anatomiques très fins. **Figure issue du domaine public, modifiée.**

INTRODUCTION

L'influx nerveux est constitué d'une suite d'*impulsions électriques*, appelées *potentiels d'action*. Lorsqu'un potentiel d'action atteint l'extrémité de l'axone d'un neurone pré-synaptique, il déclenche l'émission de molécules chimiques appelées *neurotransmetteurs*. Les neurotransmetteurs se fixent alors sur des récepteurs présents à la membrane du neurone post-synaptique, générant des courants électriques trans-membranaires locaux. Il en résulte une variation locale du potentiel électrique, appelée *potentiel post-synaptique*. Le potentiel post-synaptique se propage le long des dendrites, jusqu'à atteindre le soma : c'est *l'intégration dendritique*. L'intégration dendritique est un phénomène complexe. En effet, chaque neurone possède des milliers de synapses distribuées sur son arborisation dendritique et les potentiels post-synaptiques générés à ces synapses interagissent au cours de leur propagation. Cette intégration *spatio-temporelle* dépend principalement des propriétés physico-chimiques des dendrites. En particulier, la longueur des dendrites, leur diamètre et la composition de la membrane sont des paramètres déterminants du résultat de l'intégration dendritique [Williams and Stuart, 2002] [Stuart et al., 2016]. Le soma joue le rôle d'intégrateur final de tous les potentiels post-synaptiques provenant des dendrites. Si le potentiel électrique résultant dépasse un certain seuil alors, le neurone génère lui-même un nouveau potentiel d'action, suivant une loi du « tout ou rien ». Cette impulsion est alors transmise aux neurones suivants *via* son axone.

Comment les neurones permettent-ils l'émergence de fonctions cérébrales complexes, simplement par des échanges d'impulsions électriques et sans supervision ?

Le cerveau contient près de 100 milliards de neurones [Herculano-Houzel, 2009] qui communiquent ainsi, au sein d'une multitude de réseaux et sous-réseaux interconnectés de façon extrêmement intriquée. Malgré les progrès technologiques et des recherches intensives, la réponse à cette question conserve une grande part d'inconnu. Paradoxalement, notre intuition et notre rationalité, qui doivent au cerveau leur existence, échouent quand il s'agit de le comprendre [Singer, 2007].

Les recherches en neurosciences proposent néanmoins des théories restreintes du fonctionnement du cerveau. Une théorie de l'esprit, particulièrement populaire dans les années 60-70, est fondée sur une *métaphore cerveau/ordinateur* (ou *computationnalisme*) [Horst, 2011]. Suivant cette théorie, la pensée complexe pourrait être réduite à l'application d'algorithmes et de calculs élémentaires. Comme l'ordinateur est la concrétisation matérielle de la machine abstraite de *Turing*, le cerveau serait le support matériel de l'esprit [Notari, 2009]. Ce qui a inspiré cette métaphore est l'assimilation entre le code binaire de base de la machine de Turing et la loi du « tout ou rien » des potentiels d'action. Cette métaphore est réductrice et se heurte à des paradoxes si l'on considère les principes de l'ordinateur d'aujourd'hui [Peytavin, 1987]. Par exemple, au contraire de l'ordinateur, le cerveau ne possède pas d'unité centrale de traitement. Surtout, l'ordinateur traite de l'information codée dans chaque bit de façon conventionnelle et stricte, alors que le cerveau est un système réactif dont l'activité dépend de quantités d'impulsions électriques variables et non codantes *stricto sensu*.

D'autres théories du fonctionnement cérébral, fondées sur des bases plus biologiques et psychologiques, ont été proposées. L'une d'elles, particulièrement influente, a été proposée par *Donald Olding Hebb* en 1949 [Hebb, 1949]. Il a postulé que les fonctions cérébrales proviendraient de propriétés particulières des neurones et de leur capacité à constituer des *assemblées de neurones*. Une assemblée de neurones est une structure fonctionnelle transitoire au sein d'un réseau de neurones qui se définit par la synchronisation de l'activité des neurones qui la composent [Singer and Gray, 1995] [Harris, 2005]. Un neurone donné peut participer à différentes assemblées au cours du temps. Cependant, il peut également ne pas participer systématiquement à une assemblée donnée, d'une occurrence à l'autre. La formation des assemblées de neurones reposerait sur des processus, encore mal connus, de *coordination spatio-temporelle* de l'émission et de la réception des potentiels d'action, et sur des processus de *plasticité*. La co-activation répétée des neurones renforcerait les connexions synaptiques qui les relient. C'est le fameux principe de la « synapse de Hebb ». Ceci aurait pour conséquence de faire émerger des groupes de neurones, au sein desquels la communication serait optimisée. Dès lors, c'est toute l'assemblée qui serait activée en réponse à un stimulus donné, constituant ainsi l'unité de représentation perceptive, comportementale et cognitive [Eichenbaum, 2017]. Le cerveau serait donc un système biologique dynamique, au sens qu'il se modifie au cours du temps du simple fait de son fonctionnement. Cela implique qu'une même fonction peut être réalisée de façon différente à chaque itération. Un tel système est à l'évidence très différent d'un ordinateur.

Limites expérimentales et approches *in silico*.

La théorie des assemblées de Hebb a longtemps principalement servi d'outil conceptuel en neurosciences. Depuis les années 80-90, avec l'apparition des techniques d'enregistrement unitaires multi-électrodes, des observations expérimentales s'accumulent en faveur de cette théorie [Nicollelis et al., 1997]. Cependant, le processus de mise en place et la fonction exacte des assemblées sont toujours loin d'être compris.

Pour étudier expérimentalement les assemblées, il est nécessaire d'enregistrer plusieurs neurones simultanément, afin de détecter une éventuelle synchronisation temporelle [Reynaud-Bouret et al., 2014] [Tuleau-Malot et al., 2014]. C'est ce qui constitue la principale difficulté technique dans ce contexte. Le caractère dynamique des assemblées complique également leur étude : elles ne sont actives que transitoirement, ce qui les rend difficiles à capturer. Enfin, et surtout dans le cadre de notre problématique, l'observation des assemblées ne peut se faire que lorsqu'elles sont établies, ce qui ne permet pas d'étudier leur mise en place.

Pour permettre de dépasser les limites expérimentales, les approches *in silico* s'avèrent très prometteuses. Ce type d'approche s'appuie sur des modèles informatiques qui simulent le fonctionnement du système biologique. En travaillant sur des modèles, il est alors possible d'observer et de mieux comprendre des phénomènes difficiles, voire impossibles à étudier *in vitro* ou *in vivo*. Dans le contexte des assemblées, il s'agit notamment du seul moyen de tester des hypothèses relatives à leur mise en place. Cette stratégie *in silico* fait l'objet des

INTRODUCTION

neurosciences computationnelles. C'est dans ce contexte que se situe de cette thèse, qui se concentre sur la modélisation d'un neurone unique en termes d'entrées/sorties.

Le fonctionnement des assemblées de neurones implique naturellement de s'intéresser à la dynamique des réseaux de neurones. Néanmoins, cette dynamique complexe, à l'échelle des réseaux, *émerge* des propriétés particulières des neurones individuels. S'intéresser à l'échelle du neurone individuel est donc une étape primordiale, préalable à la mise en réseau. C'est ce qui constitue le cœur de cette thèse.

Il existe déjà de nombreux modèles informatiques de neurones uniques ayant des caractéristiques très variées. On les distingue notamment sur la base de leur niveau d'abstraction, et des objectifs de la modélisation. À un niveau très abstrait, on trouve par exemple le modèle de *McCulloch et Pitts* (1943) [McCulloch and Pitts, 1943]. Le neurone est modélisé comme une unité élémentaire de calcul logique qui traite des entrées binaires et renvoie une sortie binaire également. La représentation de l'information sous forme de valeur 0 ou 1, renvoie aux potentiels d'action échangés par les neurones biologiques. Malgré cette analogie, ce modèle est peu représentatif de la véritable fonction du neurone biologique, au vu des connaissances actuelles. En particulier, on sait que l'*intégration dendritique* et les « effets mémoire » du soma sont déterminants dans la fonction du neurone, et ceux-ci sont totalement ignorés dans le modèle de McCulloch et Pitts. Ce type de modèle, extrêmement simpliste, n'est donc pas adapté à l'étude profonde des mécanismes neuronaux. Cependant, ces modèles *bio-inspirés*, qui focalisent surtout sur les propriétés calculatoires des neurones, ont d'autres applications. Ils sont actuellement exploités en grands réseaux, pour l'intelligence artificielle.

À l'opposé, il existe des modèles de neurones que l'on dit *biologiquement plausibles*, très détaillés, capables de reproduire avec précision le comportement des neurones biologiques. Le plus utilisé en neurosciences computationnelles est le modèle *Hodgkin-Huxley* [Hodgkin and Huxley, 1952d] qui focalise sur les mécanismes moléculaires à l'origine de l'activité bioélectrique des neurones. Ce modèle est régi par un système d'équations différentielles. C'est un modèle très intéressant pour étudier les principes qui sous-tendent le fonctionnement des neurones. En particulier, il peut être combiné avec un autre modèle du même type, issu de la *théorie des câbles* [Rall, 1962] [Markram, 2006]. Cette théorie décrit la propagation du signal dans les dendrites et permet ainsi de prendre en compte l'impact de la morphologie du neurone sur sa fonction. On retrouve notamment ces modèles au cœur du *Blue Brain Project* [EPFL, 2018], qui a pour ambition de simuler un cerveau complet, de manière détaillée à toutes les échelles biologiques. En contrepartie de cette potentialité, la complexité intrinsèque de ce type de modèles nécessite d'immenses ressources de calculs, en particulier lorsqu'il s'agit de passer à l'échelle des réseaux. De plus, le grand nombre de paramètres impliqués réduit leur pouvoir explicatif.

Il n'y a pas de frontière stricte entre les modèles que l'on dit « biologiquement plausibles », très détaillés, et les modèles « bio-inspirés », très idéalisés (voir Chapitre 3) : le niveau d'abstraction idéal dépend des objectifs de la modélisation et ce n'est pas le niveau de détails qui fait la qualité du modèle [Gerstner and Naud, 2009].

Objectif de la thèse.

Pour la communauté qui s'intéresse aux assemblées de neurones, dont *Franck Grammont* (co-encadrant de cette thèse), aucun des modèles de neurones existants n'est satisfaisant pour permettre l'étude des assemblées de neurones et surtout l'étude de leur mise en place.

L'étude des assemblées nécessite de tester des hypothèses sur de grands réseaux puisqu'elles émergent de l'activité collective des neurones. Malheureusement, lorsque les modèles existants ont un niveau d'abstraction suffisant pour permettre un passage à l'échelle, il s'avèrent trop incomplets. Effectivement, la plupart des modèles suffisamment abstraits sont ponctuels, c'est-à-dire qu'ils représentent le neurone comme un point, sans prendre en compte sa morphologie caractéristique. La morphologie des neurones est pourtant déterminante dans leur fonction. En particulier, les caractéristiques physico-chimiques des dendrites influent largement sur l'intégration spatio-temporelle des potentiels post-synaptiques. La puissance des potentiels post-synaptiques est modulée lors de la propagation dans les dendrites, des synapses au soma. En agissant sur les potentiels post-synaptiques, les dendrites conditionnent l'émission des potentiels d'action. De plus, la distribution des synapses sur l'arborisation dendritique impose, de fait, des temps de transmission variables suivant leur localisation par rapport au soma. La morphologie des neurones joue ainsi un rôle crucial dans la coordination temporelle de l'activité neuronale. Puisque les assemblées de neurones sont caractérisées par ce type de processus, il est indispensable de prendre en compte leur morphologie dans ce contexte [Harris, 2005] [Stuart et al., 2016].

Dans cette thèse, notre objectif est de définir un nouveau modèle informatique de neurone unique, adapté à l'étude *in silico* des assemblées. Finalement, nous devons définir un modèle qui doit être aussi crédible que possible, en y intégrant notamment des dendrites, pour être pertinent dans le contexte des assemblées de neurones. Cependant, ce modèle doit être suffisamment abstrait pour permettre, à terme, un passage aisé à l'échelle des réseaux. Le niveau d'abstraction requis est également contraint par notre approche qui prend le parti des *méthodes formelles*, en vue de faciliter les raisonnements à l'échelle des assemblées. Nous portons donc une attention particulière aux propriétés algébriques issues de nos définitions.

Originalité de notre approche et contribution.

Les neurosciences computationnelles sont essentiellement fondées sur la simulation numérique ou l'analyse [Baladron et al., 2012] [Delarue et al., 2015]. En ce sens, notre approche est originale puisqu'elle n'est pas centrée sur ce type de techniques. Nous adoptons une démarche plus formelle qui consiste à mettre en lumière des propriétés algébriques intéressantes pour l'utilisation, à terme, des méthodes formelles, ouvrant la porte à du *raisonnement assisté par ordinateur*.

Les méthodes formelles en informatique rassemblent différentes techniques qui permettent de *raisonner* rigoureusement sur un système, grâce à une *logique mathématique*. Le raisonnement s'appuie sur une *description mathématique formelle* du système. Les méthodes formelles sont principalement utilisées pour vérifier l'absence de bugs dans des logiciels critiques, c'est-à-dire pour assurer leur validité par rapport à des spécifications données. Ce-

INTRODUCTION

pendant, un grand nombre de recherches en informatique consistent à élargir leurs champs d'application pratique. En particulier, les méthodes formelles sont exploitées, avec succès, pour l'étude des systèmes biologiques dynamiques [Snoussi, 1989] [Bernot et al., 2004] [Heiner and Koch, 2004] [Fages et al., 2004] [Calzone et al., 2006a]. Ce type d'approche permet de prouver, automatiquement ou de manière assistée par ordinateur, que la modélisation d'un système satisfait des propriétés biologiques d'intérêt, à partir de la transcription formelle d'hypothèses. L'implication est alors double : la preuve permet de sélectionner les modèles cohérents avec la biologie pour identifier les paramètres du système et ensuite d'orienter les plans d'expériences afin de préciser la modélisation. Ce type d'approche est surtout appliqué aux réseaux génétiques et métaboliques. En revanche, peu d'applications sont recensées en neurosciences.

L'application des méthodes formelles nécessite, avant tout, une description rigoureuse du système d'étude, au sens mathématique du terme. Les méthodes formelles ne manipulent pas des valeurs, mais des symboles. La transition du rigoureux vers le formel requiert donc l'introduction de *symboles* dans la description de chaque objet. Tout ce qui peut agir sur le système doit également avoir un symbole et un domaine de définition qui spécifie les valeurs qu'il peut prendre. Cette description symbolique doit s'accompagner de propriétés mathématiques permettant de lier les symboles entre eux. Ces *propriétés remarquables* peuvent donner lieu à des *axiomes*, constituant la base des raisonnements assistés par ordinateur. Finalement, une approche formelle nécessite, en amont, une vision algébrique du système. C'est avant toutes choses notre première préoccupation, tout en restant biologiquement pertinent pour l'objectif d'étude des assemblées.

Dans cette thèse, nous proposons donc un modèle de neurone formel adapté à l'étude des assemblées. Il a la particularité d'intégrer des dendrites, à un niveau abstrait. Nous fournissons une définition rigoureuse de notre modèle et mettons en évidence ses propriétés remarquables. De plus, nous montrons *a posteriori* l'importance de prendre en compte les dendrites lorsque l'on considère la fonction d'entrée/sortie des neurones.

La modélisation du soma du neurone induit usuellement, malheureusement, une équation différentielle lorsque l'on souhaite préserver l'essence de son fonctionnement biologique. Ceci impose d'adopter également une vision analytique, qui complique la transition vers le formel. Néanmoins, nous établissons des propriétés algébriques relatives à l'arborescence dendritique. En particulier, nous prouvons qu'il est possible de réduire fortement la structure des neurones tout en préservant leur fonction d'entrée/sortie. Cette propriété met en évidence des *classes d'équivalence* dont on sait déterminer un *représentant canonique*. Un résultat essentiel de cette thèse découle de cette propriété : l'arborescence dendritique ne contribue à la fonction d'entrée/sortie des neurones que par l'apport de *délais* entre les synapses et le soma, dans le cas d'une propagation dendritique passive. L'ajout de délais dans les modèles de neurones permet de représenter une intégration dendritique abstraite, sans représenter explicitement la morphologie.

En s'appuyant sur la *théorie des catégories*, nous établissons des propriétés algébriques

plus fines. Nous démontrons principalement qu'une notion de morphisme entre neurones permet de mieux comprendre l'intégration dendritique et les comportements d'entrée/sortie, au travers d'un pré-ordre sur l'ensemble des neurones. Ce type de propriété remarquable pourrait participer aux règles de base du raisonnement formel.

Faute de produire une description totalement axiomatique formelle (à cause de l'équation différentielle au soma), on introduit en partant de la définition rigoureuse du modèle, une *discrétisation temporelle* permettant un encodage en *Lustre*. Lustre est un langage de programmation formel qui bénéficie d'outils de *model checking* dédiés. Les preuves par model checking sur notre modèle de neurone encodé en Lustre sont limitées. Cependant, la faisabilité des preuves automatiques met en évidence que nos définitions sont bien choisies. Le cadre de modélisation que nous proposons constitue finalement un premier pas vers la formalisation complète des neurones.

Les travaux présentés dans ce manuscrit sont le résultat d'une interaction pluridisciplinaire à l'interface entre l'informatique théorique, les mathématiques élémentaires et les neurosciences expérimentales et théoriques. Le contexte est proposé par un neurobiologiste, qui souhaite bénéficier d'un modèle de neurone adapté à ses problématiques de recherches, et en particulier à l'étude des assemblées. Par ailleurs, la thèse est encadrée par des informaticiens dont l'objectif est double. Il s'agit, en premier lieu, de satisfaire les besoins en neurosciences, mais également de tirer profit de la modélisation pour fournir des résultats mathématiques pouvant bénéficier à l'informatique théorique. Cette thèse n'est pas le résultat d'une démarche utilitaire, mais s'inscrit davantage dans un cercle vertueux qui consiste à apporter des bénéfices potentiels aux différentes disciplines impliquées.

Guide de lecture.

Ce manuscrit est constitué de trois grandes parties. Une première partie présente le contexte pluridisciplinaire dans lequel nous nous plaçons. Une seconde partie, qui constitue le cœur de la thèse, définit le cadre de modélisation que l'on propose. Enfin, une dernière partie traite de l'implémentation au sein de ce cadre et introduit une extension possible.

La première partie se divise elle-même en trois chapitres. Le Chapitre 1 fournit les connaissances biologiques permettant de comprendre et de justifier le cadre de modélisation proposé. Il décrit tout d'abord l'objet à modéliser, c'est-à-dire la structure et la fonction du neurone, avec une insistance particulière sur les dendrites. Puis, il présente le contexte biologique de la modélisation, à savoir la théorie des assemblées et les mécanismes de plasticité dont elle dépend. Le Chapitre 2 fournit les prérequis théoriques, mathématiques et informatiques, nécessaires à la compréhension de ce manuscrit. Il s'étend surtout sur les concepts mathématiques relatifs aux fonctions, à l'algèbre générale, à la théorie des catégories, et du côté informatique, ce chapitre aborde les langages synchrones avec le cas particulier de Lustre et le model checking associé. Enfin, le Chapitre 3 situe notre travail par rapport à l'état de l'art. Il présente les principaux modèles de neurones existants et met en avant l'absence d'un modèle satisfaisant pour l'étude des assemblées de neurones. Ce dernier chapitre insiste également sur l'apport des méthodes formelles à l'étude des systèmes biologiques dynamiques,

INTRODUCTION

motivant notre approche.

La seconde partie se divise aussi en trois chapitres. Le Chapitre 4 définit le cadre de modélisation que nous proposons. Le modèle de neurone défini dans ce chapitre a la particularité de posséder des dendrites, afin de convenir à l'étude des assemblées de neurones. Le Chapitre 5 est consacré à l'étude des propriétés mathématiques du cadre de modélisation. Ce chapitre montre notamment qu'il est possible de réduire les modèles de neurones à une forme « normalisée », tout en préservant leur fonction d'entrée/sortie. L'ensemble des formes normalisées démontre, de manière surprenante, que les délais constituent l'apport essentiel de l'arborescence dendritique pour ce qui concerne cette fonction d'entrée/sortie, pour ce qui est de la propagation dendritique passive. Le Chapitre 6, fondé sur la théorie des catégories, étend les résultats du chapitre précédent. Il montre qu'il existe un « continuum » de neurones entre tout modèle et sa forme normalisée. En particulier, le niveau de sophistication de l'arborescence dendritique dépend de la proximité d'un modèle avec sa forme normalisée. Ce chapitre unifie bon nombre de propriétés algébriques des neurones. Ces propriétés sont potentiellement intéressantes en vue d'exploiter des preuves formelles.

Une dernière partie comprend également trois chapitres. Le Chapitre 7 traite de la discrétisation temporelle du cadre de modélisation proposé dans cette thèse. C'est une étape préalable à l'implémentation. Le Chapitre 8 propose ainsi une implémentation des neurones en Lustre. Lustre est un langage de programmation synchrone qui a l'avantage de permettre l'encodage de propriétés vérifiables par model checking. Un exemple simple est présenté comme preuve de concept. De plus, dans ce chapitre, l'importance des dendrites dans le modèle est mise en évidence par simulation en Lustre. Enfin, le Chapitre 9 propose une extension du cadre de modélisation en intégrant la plasticité synaptique. C'est un mécanisme essentiel dans le contexte des assemblées, qui ouvre par ailleurs des perspectives intéressantes pour le domaine de l'apprentissage automatique.

Première partie

CONTEXTE PLURIDISCIPLINAIRE

Cette première partie a pour objectif de présenter le contexte dans lequel se situe cette thèse. Il s'agit d'un contexte à l'interface de la biologie, des mathématiques et de l'informatique. S'agissant d'une thèse en informatique, les chapitres qui suivent s'attachent surtout à fournir les prérequis biologiques et mathématiques au lecteur non spécialiste. Concernant les prérequis biologiques, l'objectif n'est pas de donner une description détaillée de la biologie du neurone, mais simplement de fournir les bases biologiques et biophysiques pour comprendre sa modélisation et les hypothèses auxquelles elle recourt.

Nous présentons ensuite l'état de l'art dans les domaines de la modélisation de neurones pour nous positionner par rapport aux modèles existants, en insistant sur l'apport des méthodes formelles à la modélisation des systèmes biologiques.



CHAPITRE 1

PRÉREQUIS BIOLOGIQUES

1.1 Le neurone : unité élémentaire du système nerveux

Alors qu'il était traditionnellement admis que le système nerveux était un maillage fibreux ininterrompu, *Santiago Ramón y Cajal* a mis en évidence, à la fin du 19^{ième} siècle, l'existence de cellules distinctes [De Carlos and Borrell, 2007]. Le neurone a ainsi été désigné comme l'unité structurelle et fonctionnelle de base du système nerveux.

1.1.1 Une structure adaptée à sa fonction

Le neurone est une cellule dont la fonction principale est l'intégration, la genèse et la transmission de l'*influx nerveux*. Il possède une morphologie particulière qui reflète sa fonction. Il est principalement constitué d'un *corps cellulaire*, *péricaryon* ou *soma*, dont émergent deux types de prolongements : les *dendrites* et l'*axone* (Figure 1.1).

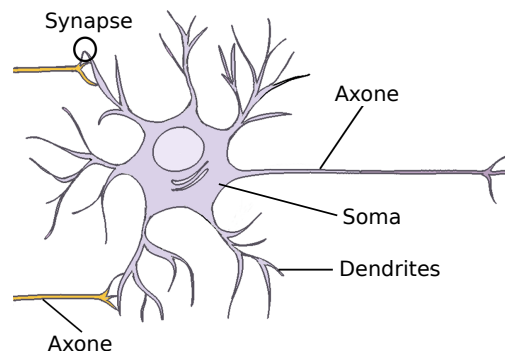


FIGURE 1.1 – Structure schématique d'un neurone archétypique. La synapse est la zone précise de connexion entre l'axone d'un neurone (dit pré-synaptique) et les dendrites d'un autre neurone (dit post-synaptique).

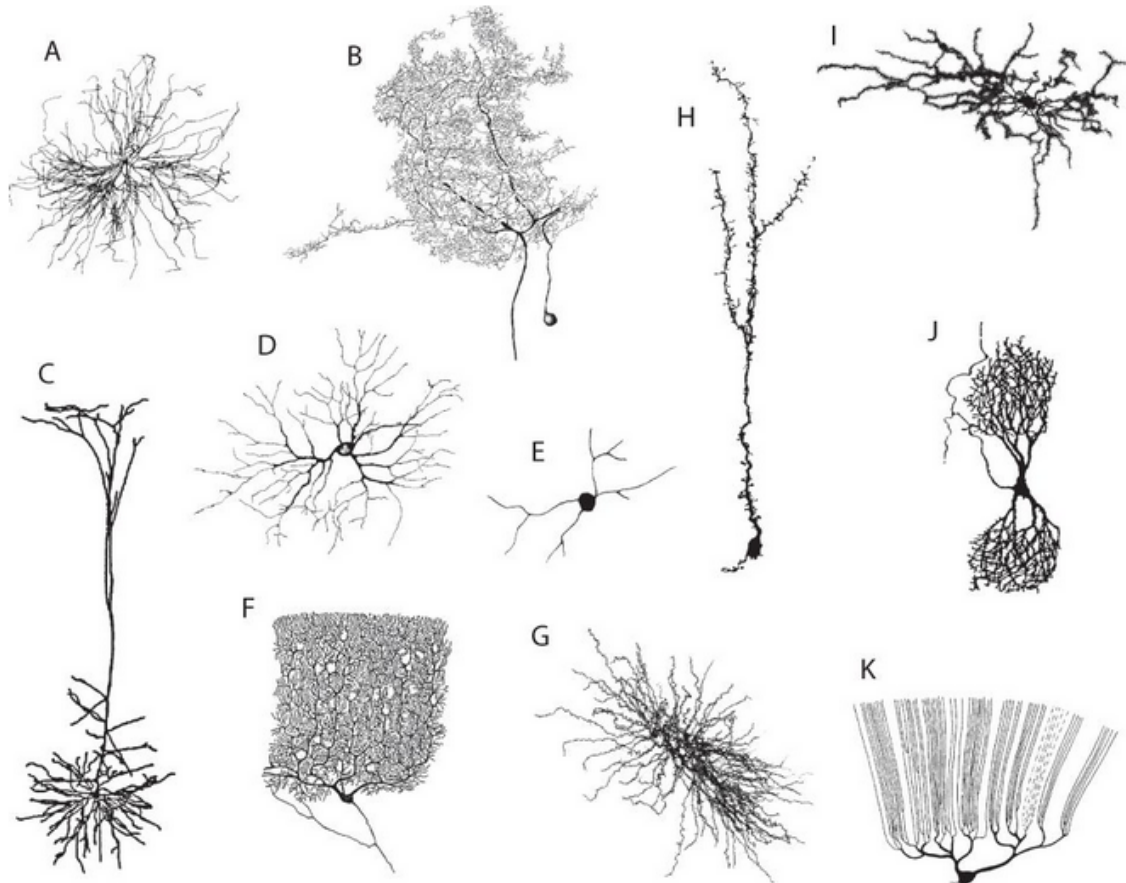


FIGURE 1.2 – Variabilité structurelle des neurones. (A) Moto-neurone de chat (B) Neurone du ganglion méso-thoracique de criquet (C) Neurone pyramidal de la couche 5 de néocortex de rat (D) Neurone ganglionnaire de rétine de rat (E) Neurone amacrine de rétine de salamandre (F) Neurone de Purkinje de cervelet humain (G) Neurone thalamique de rat (H) Neurone granulaire olfactif de souris (I) Neurone de projection épineux du striatum de rat (J) Neurone du noyau de Burdach humain (K) Neurone de Purkinje de poisson. *Figure extraite de [Stuart et al., 2016].*

Le soma est le lieu des fonctions vitales du neurone. Il contient notamment le noyau cellulaire, d'où son autre nom péricaryon, qui signifie « autour du noyau ». Il contient également d'autres organites nécessaires au fonctionnement cellulaire ordinaire (appareil de Golgi, réticulum endoplasmique, mitochondries, *etc.*). Le soma des plus grands neurones peut mesurer jusqu'à 0,1 mm de diamètre.

Les dendrites sont des prolongements du corps cellulaire. Elles servent à la réception et à la conduction des influx nerveux provenant d'autres neurones. Ce sont des structures qui peuvent être complexes et très ramifiées. Elles s'affinent généralement en s'approchant des extrémités, produisant un aspect arborescent, par dichotomies successives. Elles peuvent être recouvertes de petites protubérances, les *épines dendritiques*, qui concentrent les synapses. Nous décrirons plus précisément les dendrites dans la suite de cette section.

L'axone est un prolongement unique, dont la longueur peut varier de quelques centaines de

1.1. LE NEURONE : UNITÉ ÉLÉMENTAIRE DU SYSTÈME NERVEUX

microns à plus d'un mètre. Son rôle est de transmettre l'influx nerveux sous forme bioélectrique. L'axone possède une arborisation terminale permettant une transmission du même influx nerveux à plusieurs autres neurones. La plupart des axones des neurones du système nerveux humain sont entourés d'une *gaine de myéline* [Baumann and Pham-Dinh, 2001]. La myéline est une substance principalement lipidique, qui isole l'axone du milieu extracellulaire. La gaine de myéline provient d'autres types cellulaires (les *cellules de Schwann* dans le système nerveux périphérique et les *oligodendrocytes* dans le système nerveux central) qui s'enroulent autour de l'axone. Son rôle est d'isoler l'axone et d'augmenter la vitesse de propagation de l'influx nerveux.

Les neurones présentent une très grande variabilité structurelle et fonctionnelle (Figure 1.2). Il existe effectivement des centaines de types neuronaux [Masland, 2004] [Nelson et al., 2006] [Stuart et al., 2016]. L'identification des types neuronaux repose généralement sur le principe fondamental selon lequel des structures différentes (incluant morphologie et expression de protéines importantes) correspondent à des fonctions différentes [Masland, 2004]. Cependant, il n'y a pas de consensus sur une classification complète des types de neurones [Markram et al., 2015].

1.1.2 La perméabilité membranaire à l'origine de l'activité électrique

L'influx nerveux est le substrat de la communication entre les neurones. Il s'agit d'un signal constitué d'impulsions électriques appelées *potentiels d'action* (Figure 1.3). Les potentiels d'actions sont générés par les neurones au niveau du *cône d'implantation* de l'axone, à la jonction avec le soma. Ils sont causés par une modification de la perméabilité ionique de la membrane qui sépare l'intérieur du neurone du milieu extracellulaire. Ceci produit des courants électriques trans-membranaires à l'origine des potentiels d'action.



FIGURE 1.3 – *Enregistrement électrophysiologique du signal électrique produit par un neurone cortical pyramidal. L'encadré rouge désigne un potentiel d'action particulier. Figure extraite de [Rossant et al., 2011].*

Potentiel de repos

Au repos, la membrane du neurone est *polarisée*. C'est-à-dire que les ions sont distribués inégalement de part et d'autre de la membrane. Par conséquent, les ions étant chargés, il se crée une *différence de potentiel* électrique entre l'intérieur et l'extérieur du neurone (Figure 1.4). La face interne de la membrane est chargée négativement par rapport à sa face extracellulaire. Par convention, on donne la charge de 0 millivolts (mV) à la face extracellulaire ce

CHAPITRE 1. PRÉREQUIS BIOLOGIQUES

qui fait que la face interne est à -70 mV par rapport à l'autre : cette valeur correspond au *potentiel de repos de la membrane*. Pour mesurer cette grandeur expérimentalement, une électrode de mesure est placée au niveau de la surface interne de la membrane, alors qu'une électrode de référence est placée à la surface externe ("enregistrement intracellulaire").

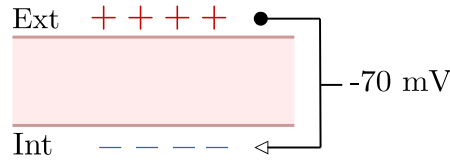


FIGURE 1.4 – *Potentiel de repos.* Au repos, la face intracellulaire (*Int*) de la membrane est chargée négativement par rapport à la face extracellulaire (*Ext*). La membrane est représentée en rose.

La membrane est une bicouche lipidique, imperméable aux ions. Cependant, des protéines trans-membranaires sont enchâssées dans la membrane. Elles forment des canaux qui permettent le passage sélectif de certains ions à travers la membrane. Les mouvements d'ions (courants) à travers ces *canaux ioniques* sont dictés par le *gradient électrochimique*. Ce gradient se divise en deux composantes : le gradient électrique (qui correspond à une variation de charge) et le gradient chimique (qui correspond à une variation de concentration). Ces deux composantes peuvent éventuellement s'opposer.

Lorsque la membrane est perméable à un seul ion, le potentiel de membrane se stabilise à une valeur propre à chaque type d'ion. Cette valeur est appelée le *potentiel d'inversion* de l'ion (noté E_{ion}). Il est déterminé par l'*équation de Nernst* :

$$E_{ion} = \frac{RT}{z_{ion}F} \ln \frac{[ion]_e}{[ion]_i}$$

où R est la constante des gaz parfaits, T est la température en kelvin, z est la charge de l'ion, F est la constante de Faraday, $[ion]_e$ et $[ion]_i$ sont respectivement les concentrations extracellulaire et intracellulaire de l'ion considéré. Si le potentiel d'inversion de l'ion est inférieur au potentiel de membrane, le flux net d'ions est sortant et inversement (Table 1.1).

| Ion | $[ion]_i$ (mM/L) | $[ion]_e$ (mM/L) | E_{ion} (mV) | Flux net au repos |
|-----------|------------------|------------------|----------------|-------------------|
| K^+ | 150 | 5.5 | -86 | Sortant |
| Na^+ | 15 | 150 | +60 | Entrant |
| Ca^{2+} | 0.0001 | 2 | +180 | Entrant |
| Cl^- | 10 | 125 | -66 | Équilibre |

TABLE 1.1 – *Potentiel d'inversion des principaux ions. Données typiques d'un neurone de mammifère, partiellement extraites de [Pritchard et al., 2002].*

Au repos, il se trouve que peu de canaux perméables au sodium, au calcium et au chlore sont ouverts. Il se produit donc principalement une sortie de potassium dans ces conditions. Pour maintenir le potentiel de repos à une valeur stable, la fuite de potassium doit être compensée. Pour cela, il existe des *pompes ioniques* qui permettent de transporter des ions

1.1. LE NEURONE : UNITÉ ÉLÉMENTAIRE DU SYSTÈME NERVEUX

contre leur gradient de concentration, en utilisant l'énergie issue de la dégradation de l'ATP. La pompe Na/K échange ainsi deux K^+ extracellulaires contre trois Na^+ intracellulaires.

Potentiel d'action

La capacité des neurones à générer des potentiels d'action provient d'une forte concentration de canaux ioniques particuliers, au niveau du cône d'implantation de l'axone. Ce sont des canaux *voltage-dépendants*, c'est-à-dire qu'ils s'ouvrent lorsque le potentiel de membrane franchit un *seuil* donné (*le seuil de dépolarisation*).

Lorsqu'un neurone reçoit des potentiels d'action provenant d'autres neurones, son potentiel de membrane peut être augmenté. Nous verrons, juste après, comment ceci se produit. Cette modification, si elle est suffisante, provoque l'ouverture des canaux voltage-dépendants, et en particulier l'ouverture de canaux sodiques. Le sodium, qui a tendance à entrer dans le neurone (Table 1.1), rend soudainement le potentiel de membrane « moins négatif » qu'au repos. La membrane se trouve alors *dépolarisée*. L'augmentation du potentiel est telle qu'il s'inverse par rapport au repos. Les canaux sodiques sont ensuite rapidement inactivés, ne laissant plus entrer d'ions Na^+ . À ce moment-là, des canaux potassiques voltage-dépendants s'ouvrent, provoquant un courant sortant d'ions K^+ . Par conséquent, le potentiel de membrane retourne vers sa valeur de repos. Il s'en suit une légère *hyperpolarisation* qui provient d'un excès de K^+ sortant. Le potentiel de repos est ensuite de nouveau maintenu par la pompe Na/K .

C'est cette variation, forte et brève (de l'ordre d'une à quelques millisecondes), du potentiel de membrane, qui constitue le potentiel d'action (Figure 1.5). Le potentiel d'action répond à une loi du « tout ou rien ». Si le neurone est suffisamment stimulé pour déclencher l'ouverture des canaux voltage-dépendants, alors un potentiel d'action est généré, sinon rien.

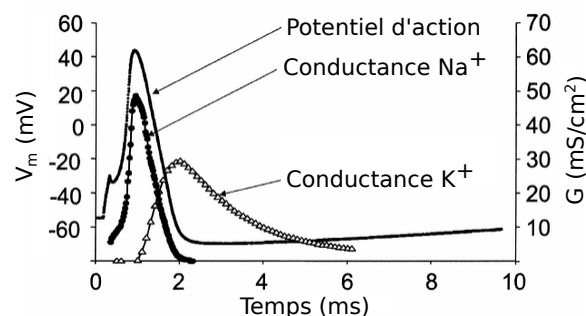


FIGURE 1.5 – Potentiel d'action et courants ioniques associés. La conductance caractérise la perméabilité de la membrane à un type d'ion spécifique. V_m est le potentiel de membrane. **Figure traduite à partir de Bezanilla et al. (2006) [Bezanilla, 2006].**

Lorsque les canaux Na^+ sont inactivés, il ne peuvent pas s'ouvrir, même s'ils sont stimulés par le voltage. Le neurone ne peut donc pas générer de nouveaux potentiels d'action pendant cette période. C'est ce que l'on appelle la *période réfractaire absolue*. La période réfractaire

CHAPITRE 1. PRÉREQUIS BIOLOGIQUES

absolue est suivie d'une *période réfractaire relative*. Durant cette période, le neurone doit être stimulé plus fortement pour déclencher un potentiel d'action.

Les potentiels d'action sont *stéréotypés*, c'est-à-dire qu'ils ont, à peu près, toujours la même forme. Ce n'est donc ni l'amplitude ni la durée du potentiel d'action qui peut définir l'intensité de l'influx nerveux. C'est uniquement la présence ou l'absence de potentiels d'action qui importe. Classiquement, il était admis que c'est exclusivement la fréquence des potentiels d'action qui code l'intensité [Adrian, 1928] [bulletin officiel, 2011]. Il existe également d'autres théories complémentaires du codage neuronal, et de plus en plus d'observations viennent appuyer un codage par le temps *précis* des potentiels d'action (voir Section 1.3).

1.1.3 La transmission de l'influx nerveux

Conduction du potentiel d'action dans l'axone

Une fois généré au cône d'implantation, le potentiel d'action est conduit le long de l'axone, avant d'être transmis à d'autres neurones. La vitesse de conduction dépend principalement du diamètre de l'axone et de la myélinisation.

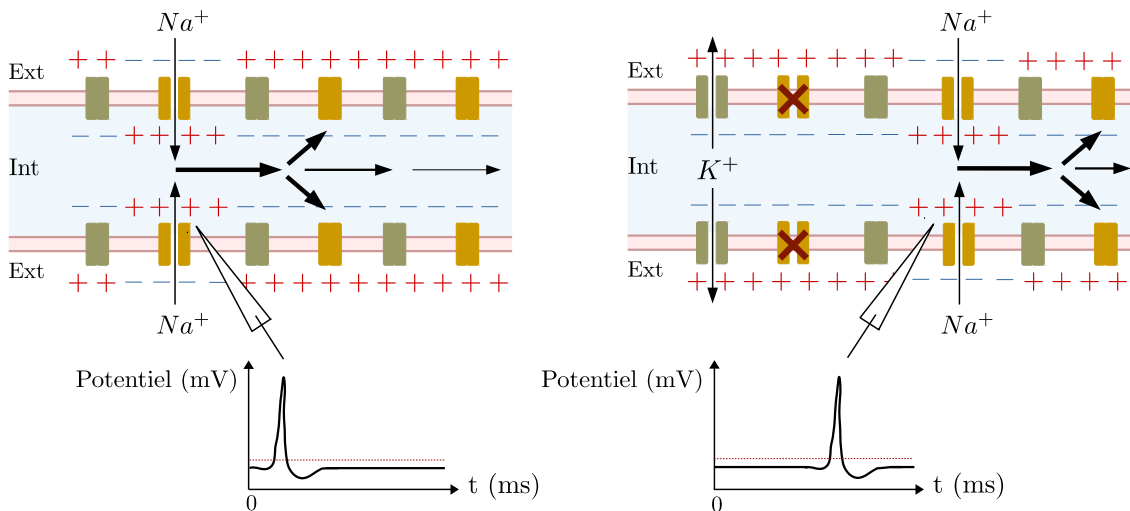


FIGURE 1.6 – Propagation du potentiel d'action dans l'axone. La genèse d'un potentiel d'action (entrée de Na^+) active des canaux voltages-dépendants adjacents qui génèrent à leur tour un potentiel d'action. Quelle que soit la zone d'enregistrement sur l'axone, le potentiel d'action conserve son amplitude originale. La croix rouge indique des canaux inactivés.

La conduction se fait sans altération du potentiel d'action. Le potentiel est en fait *re-généré* continuellement, de proche en proche. Localement, lorsque les canaux sodiques voltage-dépendants s'ouvrent, il se produit une dépolarisation. Cette dépolarisation provoque l'ouverture des canaux adjacents de même type. Un nouveau potentiel d'action est ainsi généré. Les canaux sodiques en amont s'inactivent et les canaux potassiques s'ouvrent, ce qui évite une propagation antidromique du potentiel d'action. La membrane se repolarise alors en

1.1. LE NEURONE : UNITÉ ÉLÉMENTAIRE DU SYSTÈME NERVEUX

ce point. Et le processus se répète, propageant le potentiel d'action le long de l'axone (Figure 1.6).

Lorsque les axones sont myélinisés, la membrane de l'axone est beaucoup plus isolante. Cependant, la gaine de myéline est interrompue régulièrement le long de l'axone. Les zones non myélinisées, appelées *nœuds de Ranvier*, sont très riches en canaux voltage-dépendants. Les potentiels d'action se propagent en « sautant » d'un nœud de Ranvier à l'autre : c'est la propagation *saltatoire*. La vitesse de conduction du potentiel d'action peut alors atteindre 150 m/s [Purves et al., 2001b].

Transmission synaptique

Après avoir parcouru l'axone, le potentiel d'action atteint la synapse (Figure 1.1). La synapse est la jonction entre deux neurones. C'est à cette localisation précise que le neurone *pré-synaptique* transmet l'influx nerveux au neurone *post-synaptique*. Généralement, c'est l'axone du neurone pré-synaptique qui est en contact avec les dendrites du neurone post-synaptique (synapse axo-dendritique). Cependant, il existe également des synapses axo-somatiques, axo-axoniques, dendro-dendritiques, *etc.* (plus rares et dont le fonctionnement n'est toujours pas complètement élucidé).

Il existe des synapses *électriques* et des synapses *chimiques*. Au niveau de la synapse chimique, l'activité électrique portée par les potentiels d'action n'est pas directement transmise d'un neurone à l'autre. Elle passe par la conversion en une activité chimique, qui est convertie de nouveau en activité électrique. Les neurones pré- et post-synaptiques ne sont alors pas en contact direct. Il existe un petit espace, d'une vingtaine de nanomètres, qui sépare leurs membranes respectives. Cet espace est appelé la *fente synaptique* (Figure 1.7, (b)).

L'arrivée du potentiel d'action à la terminaison pré-synaptique, active des canaux calciques voltage-dépendants (Figure 1.7, (1,2)). Le flux entrant d'ions Ca^{+2} ainsi produit, déclenche l'exocytose de vésicules (Figure 1.7, (3)). Elles contiennent de petites molécules, appelées *neurotransmetteurs*, qui sont alors déversées dans la fente synaptique. Les neurotransmetteurs peuvent se fixer sur des récepteurs présents à la membrane du neurone post-synaptique (Figure 1.7, (4)). Leur fixation a pour conséquence de modifier localement la perméabilité membranaire, directement ou de manière indirecte. Il existe, en effet, deux types de récepteurs : les *récepteurs-canaux (ionotropiques)* et les *récepteurs couplés aux protéines G (métabotropiques)*. Les récepteurs-canaux sont des canaux ioniques, qui s'activent lorsqu'un ligand spécifique s'y lie. Dans l'autre cas, le récepteur active des intermédiaires qui, à leur tour, activent des canaux ioniques. L'ouverture des canaux, quels qu'ils soient, génère des flux d'ions à travers la membrane du neurone post-synaptique (Figure 1.7, (5)). Ceci fait alors varier son potentiel de membrane. Cette perturbation locale est appelée *potentiel post-synaptique*.

Il existe des synapses *excitatrices* et des synapses *inhibitrices*, en fonction des neurotransmetteurs impliqués et des ions qui traversent la membrane à la suite de leur action. La

CHAPITRE 1. PRÉREQUIS BIOLOGIQUES

fixation des neurotransmetteurs au niveau des synapses excitatrices provoque des courants ioniques dépolarisants (flux entrant de Na^+ sous l'action du glutamate, par exemple). Par conséquent, le potentiel de membrane augmente : on parle de *potentiel post-synaptique exciteur*. La dénomination « excitation » provient du fait qu'elle facilite, *in fine*, l'activation de canaux voltage-dépendants au niveau du cône d'implantation de l'axone, et donc la genèse des potentiels d'action. Au contraire, la fixation des neurotransmetteurs au niveau des synapses inhibitrices provoque des courants ioniques hyperpolarisants (flux entrant de Cl^- sous l'action du GABA, par exemple). On parle alors de *potentiel post-synaptique inhibiteur*. L'inhibition, en abaissant le potentiel de membrane, rend le neurone plus difficile à dépolariser jusqu'au seuil de dépolarisation.

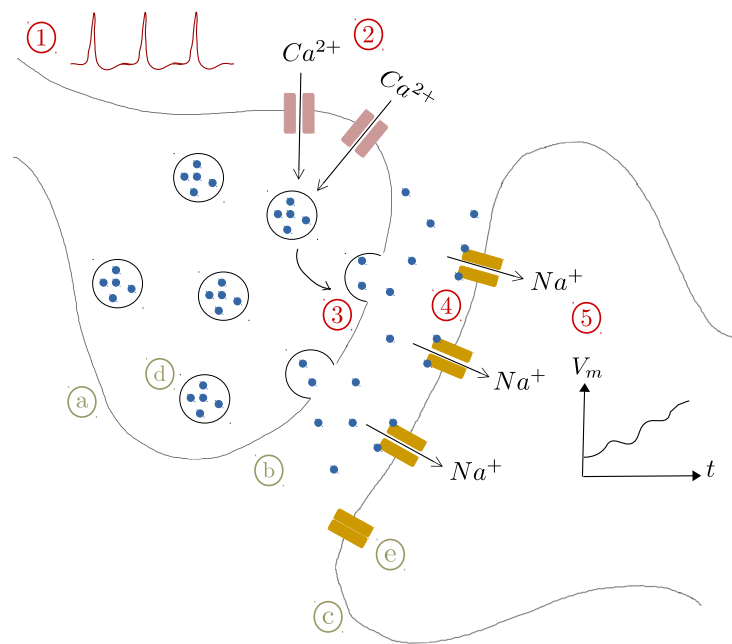


FIGURE 1.7 – Représentation schématique d'une synapse excitatrice. (1) Arrivée des potentiels d'action (2) Courants calciques (3) Exocytose des vésicules (4) Fixation des neurotransmetteurs aux récepteurs (5) Courants dépolarisants et sommation temporelle des potentiels post-synaptiques. (a) Neurone pré-synaptique (b) Fente synaptique (c) Neurone post-synaptique (d) Vésicule de neurotransmetteurs (e) Récepteur-canal ionique.

Intégration des potentiels post-synaptiques

Un unique potentiel d'action pré-synaptique, n'est généralement pas suffisant pour déclencher l'émission d'un potentiel d'action par le neurone post-synaptique. Le seuil de dépolarisation à atteindre, au cône d'implantation de l'axone, est bien plus haut que l'amplitude d'un seul potentiel post-synaptique exciteur. En fonction des neurones, il faut de quelques à plusieurs centaines de potentiels d'action pré-synaptiques « quasi-simultanés », pour permettre la genèse d'un potentiel d'action par le neurone post-synaptique.

1.1. LE NEURONE : UNITÉ ÉLÉMENTAIRE DU SYSTÈME NERVEUX

L'un des moyens d'atteindre le seuil est la *sommation temporelle* des potentiels post-synaptiques (Figure 1.8a). Au niveau d'une synapse, chaque potentiel d'action déclenche la libération d'une quantité donnée de neurotransmetteurs. Ainsi, plus il y a de potentiels d'action émis par le neurone pré-synaptique, et plus le potentiel post-synaptique varie : l'effet de chaque potentiel d'action est sommé à l'effet du précédent, s'ils sont reçus dans un laps de temps suffisamment court. Il existe également un phénomène de *sommation spatiale* qui facilite l'atteinte du seuil. Des milliers de synapses sont typiquement distribuées sur l'arborisation dendritique. Les potentiels post-synaptiques, générés au niveau de toutes ces synapses, se propagent le long des dendrites. Ils peuvent ainsi se rencontrer au cours de leur propagation. Les potentiels post-synaptiques sont alors sommés : c'est la sommation spatiale (Figure 1.8b).

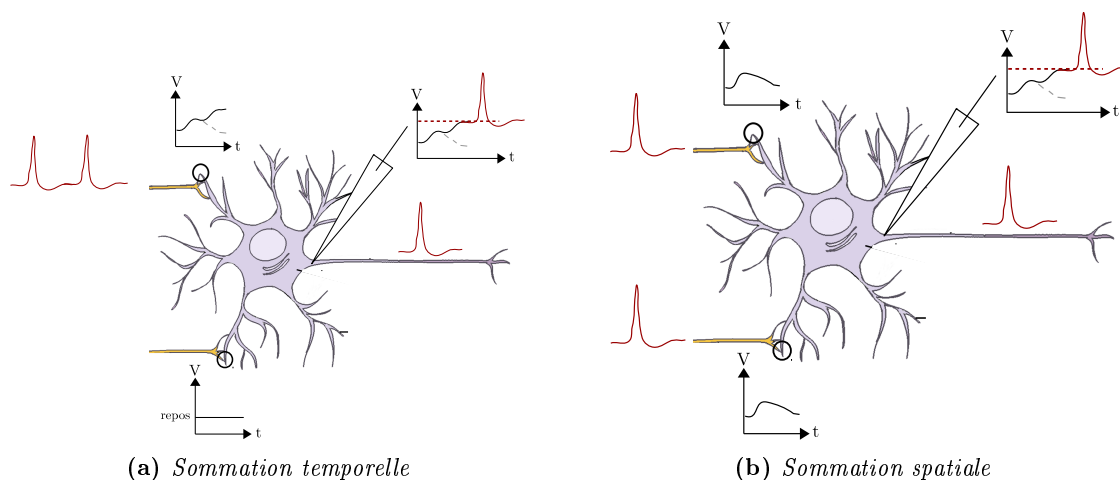


FIGURE 1.8 – *Somme spatio-temporelle*. La sommation des potentiels post-synaptiques permet au neurone d'intégrer les influx nerveux provenant de toutes les synapses. Le seuil peut être atteint par (a) sommation des potentiels post-synaptiques produits par des potentiels d'actions successifs reçus par une même synapse, ou par (b) sommation des potentiels post-synaptiques produits par des potentiels d'actions reçus à des synapses distinctes. Chaque graphique représente l'évolution locale du potentiel de membrane en fonction du temps. Le seuil de dépolarisation est représenté par des pointillés rouges.

La propagation dans les dendrites ne se fait pas de la même manière que dans l'axone. Contrairement au potentiel d'action qui est re-généré continuellement, les potentiels post-synaptiques se propagent de manière *décrémentielle*, le long des dendrites (Figure 1.9). Plus un potentiel post-synaptique s'éloigne de la synapse et plus il s'atténue. Ceci est dû à des *canaux de fuite* présents à la membrane. Ce sont des canaux qui sont toujours ouverts et qui laissent donc des ions traverser la membrane. Puisque le système tend à retourner à sa valeur de repos, le signal se trouve atténué.

À l'issue de la propagation le long des dendrites, les potentiels post-synaptiques atteignent le soma. Le soma joue le rôle d'intégrateur final de tous les influx nerveux. Si la dépolarisation est suffisante, au niveau du cône d'implantation de l'axone, les canaux voltages-dépendants sont activés. Un nouveau potentiel d'action est alors généré et transmis à d'autres neurones, *via* l'axone.

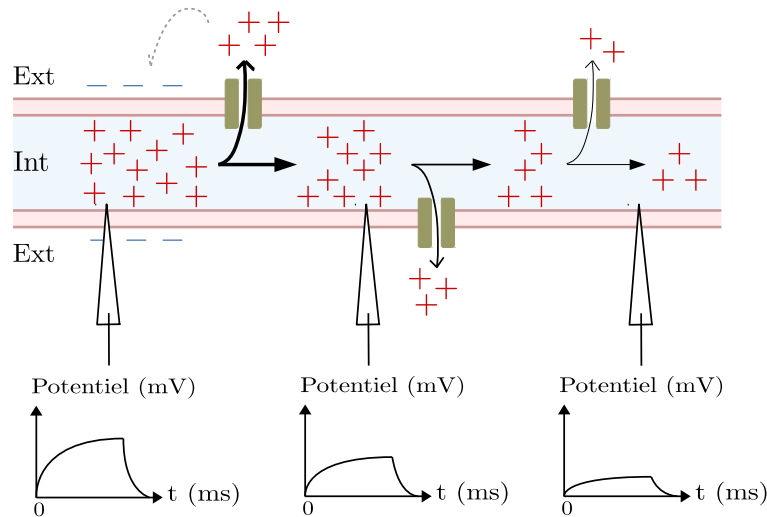


FIGURE 1.9 – Propagation décrementielle du potentiel post-synaptique dans les dendrites. La dépolarisation se propage de proche en proche, attirée par les charges négatives environnantes. Une fuite de charges fait retourner le potentiel vers sa valeur de repos. L’amplitude du potentiel post-synaptique dépend donc de la distance entre la zone de genèse et la zone d’enregistrement.

Importance de la synchronisation dans la transmission de l’influx nerveux

La transmission de l’influx nerveux, c’est-à-dire l’émission d’un nouveau potentiel d’action par le neurone post-synaptique, dépend de l’amplitude de la dépolarisation au cône d’implantation de l’axone. Comme nous venons de le voir, elle repose sur un mécanisme de sommation, à la fois spatiale et temporelle. Pour être sommés, les potentiels post-synaptiques doivent se rencontrer dans l’arborescence dendritique ou au soma. Pour se rencontrer dans les dendrites, ils doivent être temporellement coordonnés. Au soma, on pourrait facilement penser que la coordination n’est pas nécessaire. Cependant, il n’en est rien. De même qu’au niveau des dendrites, il existe une fuite importante de charges au niveau du soma. Ainsi, le potentiel de membrane tend constamment à retourner vers sa valeur de repos. Pour être sommés, les potentiels post-synaptiques doivent donc atteindre le soma de manière synchronisée. La synchronisation permet ainsi de faciliter l’atteinte du seuil et la transmission de l’influx nerveux.

1.1.4 Les dendrites et leurs propriétés

La genèse d’un potentiel d’action est le résultat d’une intégration spatio-temporelle complexe. Les dendrites sont déterminantes dans ce processus. Elles permettent non seulement de collecter les milliers d’influx nerveux que le neurone reçoit chaque seconde, mais également de moduler les potentiels post-synaptiques lors de leur propagation vers le soma.

Historique et techniques expérimentales

Les dendrites font l'objet de recherches depuis plus d'un siècle. Des observations préliminaires des dendrites sont attribuées à *Otto Deiters* en 1865 [Yuste and Tank, 1996]. Par analogie avec la structure d'un arbre, d'un système racinaire ou du système vasculaire, la structure des dendrites semblait idéale pour puiser et distribuer des nutriments. Pour cette raison, le neuroanatomiste *Camille Golgi*, a conclu que les dendrites jouaient un rôle purement nutritionnel dans la fonction du neurone [Mel, 1994]. Il a fallu attendre quelques années pour que *Ramón y Cajal* s'intéresse à la fonction des dendrites dans la propagation de l'influx nerveux. Il est à l'origine de l'idée moderne selon laquelle l'arborescence dendritique n'est pas consacrée à l'extraction de nutriments, mais à l'extraction d'information. Les dendrites permettraient notamment d'augmenter la surface de contact avec les neurones environnants. En 1889, il a constaté, au niveau des neurones de la rétine, que les dendrites étaient toujours orientées vers le monde extérieur, tandis que les axones étaient orientés vers des structures plus profondes [Yuste and Tank, 1996]. Ces observations l'ont amené à proposer la *loi de polarisation fonctionnelle* : la transmission de l'influx nerveux se fait des dendrites vers l'axone, en passant par le soma. Chaque neurone a donc une structure réceptrice (les dendrites), une structure émettrice (l'axone), et une structure de distribution (l'arborisation terminale de l'axone). Cette loi était en contradiction avec l'existence de certains neurones particuliers, pour lesquels l'axone provenait directement des dendrites. Finalement, la *loi de polarisation axipète* lui a permis de contourner ce cas particulier : le soma et les dendrites auraient une conduction axipète, c'est-à-dire qu'ils conduiraient l'influx nerveux en direction de l'axone, tandis que l'axone aurait une conduction somatofuge ou dendrifuge, en propageant l'influx vers son arborisation terminale. À l'heure actuelle, cette loi générale est toujours admise, bien que les exceptions soient importantes : l'influx peut notamment se rétro-propager vers les dendrites, jouant notamment un rôle dans l'apprentissage.

Les recherches sur les dendrites sont longtemps restées en retrait des études neurobiologiques. Le développement des techniques d'*enregistrement intracellulaire* [Brock et al., 1952], dans les années 50, a provoqué un regain d'intérêt pour les dendrites. Pour la première fois, il a été possible de s'intéresser à la physiologie interne des neurones, par le biais d'électrodes de verre qui permettent de quantifier les variations du potentiel de membrane. Dans les années 80, différentes techniques ont de nouveau révolutionné la recherche sur les dendrites. En particulier, les techniques d'imagerie ont permis de visualiser les dendrites et la distribution spatiale de l'activité bioélectrique. Grâce à la combinaison des enregistrements avec la *microscopie infra-rouge* [Dodt and Zieglgänsberger, 1990], il est devenu possible d'identifier précisément les neurones enregistrés. Par ailleurs, la synthèse d'indicateurs du calcium intracellulaire [Minta et al., 1989] a permis de localiser les changements de concentrations liés à l'activité synaptique. Deux techniques de microscopie ont largement amélioré l'imagerie sur tranches de cerveau : la *microscopie confocale* [Sheppard and Wilson, 1981] qui concentre la lumière afin d'augmenter la résolution de l'image, et la *microscopie à deux photons* [Denk et al., 1990] qui permet d'exciter des molécules fluorescentes et ainsi de visualiser

CHAPITRE 1. PRÉREQUIS BIOLOGIQUES

des localisations précises.

Un grand nombre d'études portant sur la fonction dendritique ont été réalisées *in vitro* (en dehors de l'organisme vivant). Ceci offre de nombreux avantages : il est par exemple possible de contrôler visuellement les neurones enregistrés, et l'imagerie à fluorescence produit, dans ces conditions, un bon rapport signal sur bruit. Mais en contrepartie, l'activation de synapses particulières s'avère difficile à maîtriser. La technique la plus ancienne, pour ce faire, est la stimulation par électrode. Le développement récent de l'*optogénétique* ouvre la possibilité de contrôler l'activation de neurones particuliers, en les rendant sensibles à la lumière par manipulation génétique [Petreanu et al., 2009]. Par ailleurs, la *photostimulation* permet de libérer des neurotransmetteurs (en particulier du glutamate), pour activer artificiellement des localisations précises des dendrites, avec une bonne résolution spatiale [Matsuzaki et al., 2001].

Bien que ces nouvelles méthodes aient élargi les possibilités de stimulation dendritiques, la stimulation reste néanmoins très artificielle. Ce type d'études *in vitro* permet d'obtenir des informations sur les capacités d'intégration des dendrites, mais pour connaître celles qui sont effectivement exploitées par le cerveau, des conditions plus naturelles sont nécessaires. Dans les expériences *in vitro*, la grande majorité des synapses est silencieuse, ce qui n'est pas le cas *in vivo* où le réseau neuronal est en continuelle activité. L'idéal est finalement de s'intéresser au cerveau intact. De plus, pour établir des liens de causalité entre l'intégration dendritique et le comportement, les expériences doivent être menées sur des animaux éveillés, ce qui implique de fait, de nombreuses limitations techniques et éthiques supplémentaires [Stuart and Spruston, 2015].

Malgré les progrès techniques et des recherches intensives, les propriétés et le rôle des dendrites ne sont toujours pas complètement établis. Ci-après, nous donnons une vue d'ensemble des connaissances actuelles à ce propos.

Les propriétés passives des dendrites

Longtemps considérées comme de simples « tuyaux » conduisant l'influx nerveux fidèlement, des synapses au soma, les dendrites sont en fait bien plus complexes que cela.

L'existence des potentiels post-synaptiques excitateurs et inhibiteurs a été montrée grâce aux enregistrements intracellulaires [Coombs et al., 1955]. Cette technique a également permis de localiser la genèse des potentiels d'action au niveau du cône d'implantation de l'axone [Fuortes et al., 1957]. De ces deux observations, on a pu déduire que la genèse des potentiels d'actions dépendait des contributions apportées par les potentiels post-synaptiques, au potentiel de membrane du cône d'implantation de l'axone. Ces contributions dépendent intégralement des propriétés biophysiques des dendrites que les potentiels post-synaptiques parcourent pour atteindre ce point particulier. C'est ce qu'a proposé *Wilfried Rall*, dans une modélisation mathématique pionnière des dendrites (Chapitre 3, Section 3.1). Il a ainsi mis en évidence le phénomène d'atténuation des potentiels post-synaptiques dans les dendrites (Figure 1.9). Plus précisément, il a identifié les paramètres déterminants de cette atténu-

1.1. LE NEURONE : UNITÉ ÉLÉMENTAIRE DU SYSTÈME NERVEUX

tion, qu'il a également quantifiés sous certaines conditions. Les paramètres principaux sont : la résistance membranaire, la résistance intracellulaire et la capacité de la membrane. Plus la résistance membranaire est importante, moins il y a de charges qui s'échappent, et donc, moins il y a d'atténuation. Plus la résistance intracellulaire et la capacité sont fortes, plus le courant a de difficultés pour se déplacer, et moins il peut se déplacer loin. Aussi, plus la distance parcourue est grande, et plus l'atténuation est forte. La *constante de temps*, qui est définie par le produit de la résistance membranaire par la capacité, représente la fenêtre d'intégration temporelle durant laquelle les potentiels post-synaptiques peuvent être sommés. Il s'agit donc d'un facteur critique dans l'intégration des potentiels d'action pré-synaptiques.

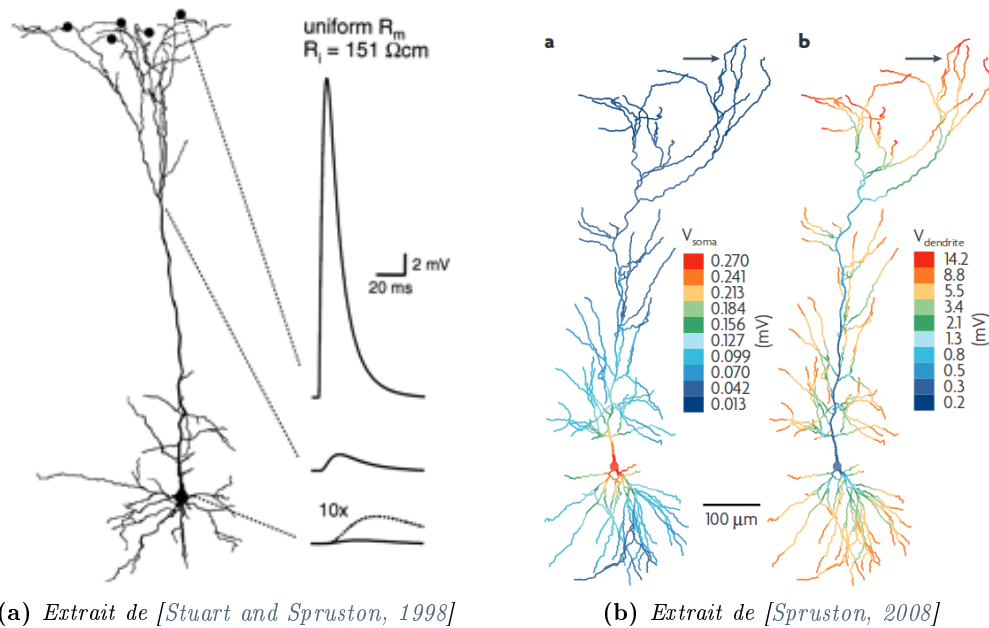


FIGURE 1.10 – Atténuation des potentiels post-synaptiques au cours de leur propagation dans les dendrites. (a) Représentation de la morphologie du modèle de neurone et enregistrement du potentiel post-synaptique simulé, à trois positions différentes : à $1000 \mu\text{m}$ du soma, à $596 \mu\text{m}$ du soma et au soma. La mesure au soma est amplifiée 10 fois. Ce modèle présente une forte résistance intracellulaire (R_i). (b) La partie a de la figure montre l'amplitude du potentiel post-synaptique au soma, pour chaque localisation de synapse sur les dendrites du modèle de neurone pyramidal. La partie b de la figure montre l'amplitude locale du potentiel post-synaptique généré par cette même simulation. L'échelle est logarithmique.

Ce phénomène de propagation décrementielle a été confirmé expérimentalement, par double enregistrement du potentiel, aux dendrites et au soma [Stuart and Spruston, 2015]. En retour, ces mesures quantitatives ont permis l'amélioration des modèles mathématiques. L'atténuation des potentiels post-synaptiques dans les dendrites est considérable, pouvant atteindre 40 fois, pour des synapses distales [Nevian et al., 2007]. Des simulations numériques prédisent même une atténuation de plusieurs centaines de fois, pour les neurones pyramidaux du néocortex [Stuart and Spruston, 1998] [Spruston, 2008] (Figure 1.10). Les potentiels générés au niveau des dendrites éloignées du soma produisent généralement des potentiels post-synaptiques d'amplitude supérieure. Cependant, ceci ne suffit pas à compenser l'atté-

CHAPITRE 1. PRÉREQUIS BIOLOGIQUES

uation [Williams and Stuart, 2002] [Spruston, 2008].

Les dendrites permettent l'intégration de tous les potentiels post-synaptiques générés aux synapses. En particulier, aux embranchements de l'arborisation, les potentiels post-synaptiques sont sommés. *Lorente de Nó* a pointé du doigt la richesse apportée par la sommation, en comparaison avec la loi du « tout ou rien », dont dépend le potentiel d'action [De Nó and Condouris, 1959].

Ces propriétés des dendrites, qui agissent sur les potentiels post-synaptiques, sont des propriétés *passives*. Les potentiels post-synaptiques se propagent simplement par interaction avec les charges environnantes. Au contraire, lorsque les potentiels d'action sont générés au cône d'implantation de l'axone, ils se propagent de manière active et régénérative le long de l'axone. Cependant, il a également été montré que les potentiels d'action peuvent aussi se rétro-propager, de manière passive, dans l'arborisation dendritique [Stuart and Spruston, 2015].

Les dendrites possèdent également des propriétés électriques *actives*. Cependant, le modèle exclusivement passif est toujours enseigné dans l'enseignement supérieur et a longtemps dominé la recherche sur les dendrites. Quoiqu'il en soit, les propriétés passives sont à la base de la fonction des dendrites et il est donc nécessaire de les prendre en compte, même lorsque l'on s'intéresse aux propriétés actives. Par ailleurs, les paramètres passifs basiques sont très difficiles à mesurer expérimentalement sur les dendrites. De plus, ils ne sont certainement pas constants le long d'un neurone donné. À l'heure actuelle, leurs valeurs ne sont généralement que des estimations à partir de simulations numériques [Stuart and Spruston, 2015]. Les propriétés passives des neurones ne sont donc toujours pas intégralement comprises et vérifiées. De plus, leur étude, qu'elle soit expérimentale ou *in silico*, reste un sujet ardu encore aujourd'hui.

Les propriétés actives des dendrites

Il a été montré expérimentalement, que les dendrites contiennent des canaux voltage-dépendants. Ainsi, elles possèdent quelques propriétés actives de nature similaire à celles de l'axone. Il existe une grande variété de canaux voltage-dépendants sur les dendrites. On y trouve surtout des canaux calciques, mais aussi sodiques et potassiques. La densité des canaux dépend du type de neurone considéré et de la localisation précise sur les dendrites [Stuart and Spruston, 2015]. Grâce à ces canaux, les dendrites sont capables de générer des *potentiels d'action dendritiques* (Figure 1.11).

Les potentiels d'action dendritiques sont différents des potentiels d'action classiques, initiés au cône d'implantation de l'axone. Contrairement aux potentiels d'action classiques, qui peuvent se propager sur de longues distances sans atténuation, les potentiels d'action dendritiques se propagent généralement de manière passive et plus lentement. De plus, ils n'ont pas la même forme, notamment, car les canaux à l'origine de leur genèse sont différents (Figure 1.11). Les potentiels d'action dendritiques sont, dans la majorité des cas,

dus à l'activation de canaux calciques. Ceux-ci sont plus larges que les potentiels d'action classiques, de type sodique. Les potentiels d'action dendritiques ont un seuil de dépolarisation plus élevé. Leur déclenchement nécessite une stimulation plus importante, de préférence synchrone et co-localisée [Williams and Stuart, 2002] [Losonczy and Magee, 2006]. Des expériences de photostimulation ont effectivement montré qu'une région dendritique localisée doit être stimulée une vingtaine de fois en quelques millisecondes, pour donner lieu à un potentiel d'action [Losonczy and Magee, 2006].

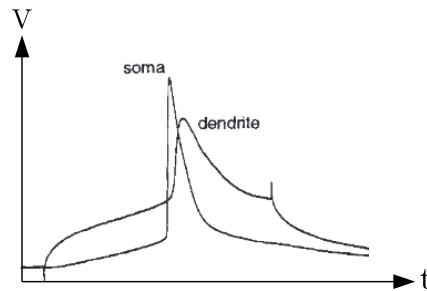


FIGURE 1.11 – Potentiels d'action somatique et dendritique, provoqués par injection locale d'un courant dépolarisant. L'enregistrement provient d'un neurone pyramidal (L5). *Figure extraite de [Stuart and Sakmann, 1994] et modifiée.*

Le rôle de ces propriétés actives n'est pas clair. Les potentiels d'action dendritiques permettraient surtout d'amplifier les potentiels post-synaptiques [Stuart and Spruston, 2015]. Même si la propagation des potentiels d'action dendritiques n'est pas optimale, leur contribution au soma pourrait permettre de compenser potentiellement l'atténuation. Par ailleurs, l'existence de canaux sodiques voltage-dépendants sur les dendrites permettrait de supporter la rétropropagation des potentiels d'action initiés à l'axone. L'efficacité de la rétropropagation dépend donc de la densité des canaux. Les potentiels rétro-propagés auraient un rôle important dans la fonction du neurone, en prenant part aux *mécanismes de plasticité*, que nous détaillerons dans la section suivante [Yuste and Tank, 1996].

Les propriétés actives des neurones permettent finalement de démultiplier les capacités des neurones, en permettant des opérations non linéaires lors de l'intégration dendritique.

Rôle des dendrites dans la fonction du neurone

La variabilité morphologique d'un type de neurone à l'autre provient principalement de la géométrie des dendrites. Les dendrites sont des structures qui peuvent être très élaborées. Elles peuvent représenter jusqu'à 99% de la surface de certains neurones.

Les dendrites varient d'un type de neurone à un autre, mais également, entre espèces (Figure 1.12). Alors que pour la plupart des organes, c'est le nombre des cellules qui augmente avec la taille de l'espèce, les neurones se mettent à l'échelle. On constate que la taille des dendrites augmente avec la taille du cerveau de l'espèce considérée. Cette mise à l'échelle préserverait notamment, voire augmenterait, le nombre de connexions par neu-

CHAPITRE 1. PRÉREQUIS BIOLOGIQUES

rone et l'architecture des microcircuits. L'augmentation de la taille des dendrites suggère une augmentation du temps de parcours et de l'atténuation des potentiels post-synaptiques, lors de leur propagation vers le soma. Dans certains cas, ceci pourrait être compensé par une augmentation du diamètre des dendrites. Cependant, l'adaptation du diamètre n'est pas automatique : certains types de neurones voient leur diamètre inchangé alors que pour d'autres, il est largement augmenté. Le type de mise à l'échelle pourrait renseigner sur le type d'intégration dendritique privilégié du neurone. La conservation du diamètre malgré l'augmentation de la taille du cerveau serait suffisante pour préserver la fonction des dendrites courtes ou avec d'importantes propriétés actives, tandis qu'une forte augmentation du diamètre permettrait de conserver le taux d'atténuation le long des longues dendrites passives. Ce dernier type d'adaptation a également été observé lors du développement de certains neurones. Ces observations suggèrent une préservation de l'efficacité relative des synapses indépendamment des changements morphologiques. La morphologie des dendrites serait donc dictée par la fonction des neurones. C'est d'ailleurs une règle générale en biologie concernant les relations structure-fonction : la fonction contraindrait la structure, au cours de l'évolution. Cependant, il n'existe pas de classification morphologique stricte. Il pourrait donc exister un « continuum » entre les différents neurones, que ce soit du point de vue de la morphologie, ou de la fonction.

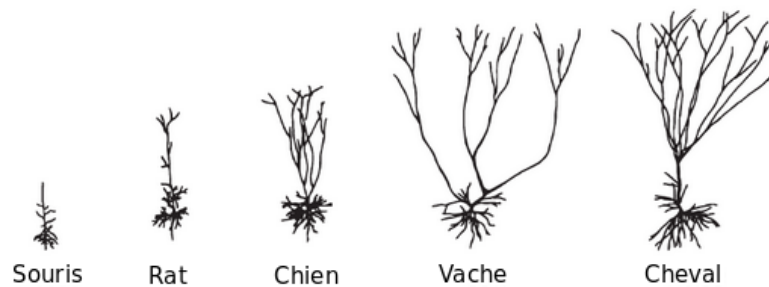


FIGURE 1.12 – *Neurones pyramidaux du néocortex, chez différentes espèces de vertébrés. Figure extraite et traduite de [Stuart et al., 2016].*

La forme des dendrites serait adaptée à la connectivité. On sait que les connexions se font sur des domaines particuliers des dendrites, suivant leur type et leur provenance (Figure 1.13). Par exemple, sur les neurones pyramidaux du CA1 (hippocampe), les synapses inhibitrices sont préférentiellement localisées à proximité du soma [Megias et al., 2001]. De plus, ces neurones reçoivent des influx nerveux sur des domaines distincts. Les dendrites proximales reçoivent principalement des influx nerveux de source locale, tandis que les localisations distales reçoivent des influx provenant de régions corticales et thalamiques distantes. La raison de cet arrangement particulier, guidé par des gradients chimiques, est actuellement inconnue. Elle suggère néanmoins que chaque source doit être intégrée différemment.

En agissant sur la contribution des potentiels post-synaptiques, les dendrites conditionnent l'intégration des influx nerveux reçus. Elles agissent ainsi sur l'atteinte du seuil de dépolarisation, et jouent donc un rôle essentiel dans la fonction du neurone.

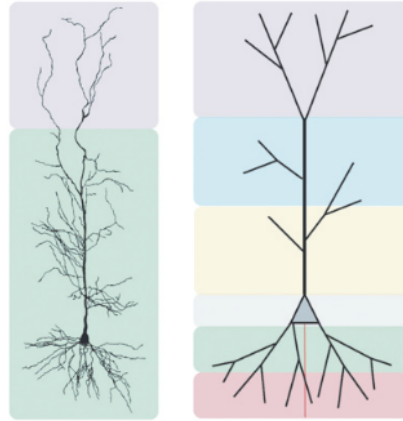


FIGURE 1.13 – *Neurone pyramidal du CA1. Sur la partie gauche de la figure, la partie apicale des dendrites (sur fond grisé) reçoit des influx excitateurs ayant des origines pré-synaptiques différentes de ceux reçus en région proximale (sur fond vert). Le schéma de droite illustre les différents domaines dendritiques qui pourraient recevoir un unique type d'influx. Des évidences s'accroissent en faveur d'une division de l'arborescence dendritique en, au moins, ces domaines, correspondant chacun à des origines pré-synaptiques distinctes. Figure extraite de [Spruston, 2008].*

Les dendrites influencent également l'émission des potentiels d'action en jouant sur un autre paramètre déterminant : la dimension temporelle. Cet aspect est peu discuté dans la littérature. À titre d'exemple, dans le livre de référence *Dendrites*, publié par *Greg Stuart, Nelson Spruston* et *Michael Häusser* en 2016, seuls quelques paragraphes sur les quelque 700 pages qu'il contient sont consacrés à l'influence des dendrites sur la dimension temporelle de l'intégration. Suivant la localisation des synapses, les dendrites induisent, de fait, un délai variable entre la genèse des potentiels post-synaptiques et leur intégration au soma. La propagation dans les dendrites est de l'ordre de $0,1 \text{ mm/ms}$ [Stuart et al., 2016]. C'est beaucoup plus lent que dans l'axone, où la vitesse de conduction du potentiel d'action varie de $0,5$ à 10 m/s pour les axones non myélinisés et jusqu'à 150 m/s en présence de myéline [Purves et al., 2001b]. Le délai entre la genèse du potentiel post-synaptique et son intégration au soma, peut ainsi atteindre 20 ms . Ce délai est inévitablement en jeu dans la synchronisation des potentiels post-synaptiques au soma et influe, par conséquent, sur l'atteinte du seuil de dépolarisation et l'émission des potentiels d'action.

Les dendrites sont des structures complexes, dont la morphologie semble intimement liée à la fonction des neurones. L'hétérogénéité de la géométrie et des propriétés des dendrites peut donner lieu à une infinité d'intégrations dendritiques possibles. Comprendre comment les neurones intègrent les signaux d'entrée qu'ils reçoivent est critique pour comprendre le fonctionnement du cerveau. En particulier, les dendrites semblent jouer un rôle dans la coordination des potentiels d'action entre les neurones. Nous verrons dans la dernière section de ce chapitre que la coordination pourrait justement être à la base des fonctions cérébrales. Cependant, le fonctionnement et le rôle des dendrites sont encore partiellement incompris. Une combinaison d'expérimentation et de théorie est certainement nécessaire au progrès.

1.2 Quelques mots sur la plasticité

Le cerveau n'est pas une structure figée. À l'évidence, le cerveau évolue lors du développement ou en cas de lésion, mais il évolue aussi, chaque seconde, tout au long de la vie d'un individu, en fonction de l'environnement et des expériences vécues. C'est ce qu'on appelle la *plasticité*. La plasticité implique des modifications à différents niveaux d'organisation.

À l'échelle des structures cérébrales, on parle de *plasticité cérébrale* : les circuits neuronaux peuvent être créés ou remaniés. À l'échelle du neurone, c'est la *plasticité neuronale* : les neurones peuvent se développer ou régresser. Il existe également de la plasticité à l'échelle subcellulaire, en particulier au niveau des synapses. C'est ce dernier type de plasticité, la *plasticité synaptique*, qui serait à l'origine de la mémoire et des apprentissages. La plasticité synaptique désigne la modification dynamique de l'efficacité des synapses, et par conséquent de la probabilité que l'influx nerveux passe d'un neurone pré- à un neurone post-synaptique. Cette modification dépend directement de l'activité bioélectrique des neurones eux-mêmes. On distingue plusieurs types de plasticité synaptique, suivant la durée de persistance des modifications et leur type.

Il existe une forme de plasticité à *court terme*, qui débute à l'échelle de la milliseconde. Elle cause des modifications de la transmission synaptique qui durent au maximum quelques minutes. On parle de *facilitation* (ou *potentialisation*) lorsqu'il s'agit d'une augmentation de l'efficacité des synapses, et de *dépression* lorsqu'il s'agit d'une diminution. Plusieurs mécanismes peuvent expliquer cette facilitation. Elle se produit notamment lorsque plusieurs potentiels d'action atteignent successivement une terminaison pré-synaptique. Il se produit alors une augmentation du calcium intracellulaire, qui persiste dans le temps. Ainsi, lorsqu'un potentiel d'action arrive peu après un autre, il bénéficie de l'entrée de calcium due au potentiel d'action précédent et par conséquent, une plus grande quantité de vésicules libèrent leurs neurotransmetteurs. Au contraire, la dépression synaptique est due à une déplétion du *pool* de neurotransmetteurs présents à la terminaison pré-synaptique. L'efficacité de la synapse est alors réduite, jusqu'à ce que de nouvelles vésicules soient disponibles. De nombreux autres mécanismes peuvent également être impliqués et provoquer le même type d'effets. La plasticité à court terme, de fait, ne peut être à l'origine des changements de comportements qui persistent pendant des semaines, des mois, voire des années.

Il existe une forme de plasticité à *long terme*, certainement à l'origine de la mémoire et des apprentissages. On parle de *LTP* (pour *Long-Term Potentiation*) pour désigner une augmentation de l'efficacité des synapses et de *LTD* (pour *Long-Term Depression*) pour désigner une diminution. La LTP a surtout été étudiée au niveau des synapses excitatrices de l'hippocampe, mais observée dans bien d'autres structures [Augustine et al., 2004], et également au niveau des synapses inhibitrices [Castillo et al., 2011] [Kullmann et al., 2012]. L'hippocampe est une structure cérébrale particulièrement importante dans la mémoire. De plus, il possède une structure typique qui permet d'étudier des tranches *in vitro*, en conservant les circuits d'intérêt intacts. En particulier, beaucoup de travaux ont été menés sur les

1.2. QUELQUES MOTS SUR LA PLASTICITÉ

synapses excitatrices qui connectent les axones des neurones pyramidaux de la région CA3 de l'hippocampe avec les dendrites des neurones pyramidaux de la région CA1. Lorsqu'un neurone du CA3 est stimulé brièvement et à haute fréquence, on peut observer de la LTP à ces synapses. Le renforcement des synapses dépend aussi de l'*état du neurone post-synaptique*. Si un neurone du CA3 est stimulé faiblement, associé à une forte dépolarisation du neurone post-synaptique du CA1, alors on peut aussi observer de la LTP bien que la stimulation seule soit trop faible pour l'induire. La LTP est également *spécifique* : lorsqu'elle est induite par une synapse, les autres ne sont pas renforcées. Une autre propriété importante de la LTP est l'*associativité*. Puisqu'une seule synapse stimulée faiblement ne peut déclencher un potentiel d'action, plusieurs synapses doivent être stimulées simultanément. Ces synapses sont alors toutes potentialisées.

La LTD se produit lorsque les synapses sont stimulées à basse fréquence pendant de longues périodes. En ayant l'effet inverse de la LTP, la LTD permet de conserver la pertinence de la LTP. Effectivement, s'il existait uniquement la LTP, les synapses atteindraient toutes une efficacité maximale. Dans ces conditions, la LTP serait insignifiante.

Finalement, la plasticité permet de renforcer les synapses qui ont permis au neurone post-synaptique d'atteindre son seuil et déprécie celles qui sont peu actives. Avant toute mesure expérimentale, cette idée de causalité a originellement été formulée par Hebb de la manière suivante : « *When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased* » [Hebb, 1949]. En termes plus modernes, elle stipule que : si un neurone A stimule un neurone B de manière répétée ou prolongée, participant à l'émission d'un potentiel d'action par le neurone B, des changements cellulaires ou moléculaires se mettent en place de telle sorte que l'efficacité de A soit renforcée vis-à-vis de son action sur B.

Traditionnellement, les études relatives à la plasticité synaptique se focalisaient sur la fréquence des stimulations électriques, permettant l'induction de LTP ou LTD. Plus récemment, des travaux se sont intéressés à l'importance des temps précis d'émission des potentiels d'action pré- et post-synaptiques [Markram et al., 1997] [Bi and Poo, 1998] [D'amour and Froemke, 2015]. Une règle générale a ainsi émergé : lorsque le neurone pré-synaptique émet un potentiel d'action *avant* le neurone post-synaptique, de la LTP est induite, mais lorsque le neurone pré-synaptique émet un potentiel d'action *après* le neurone post-synaptique, de la LTD est induite. L'amplitude de la plasticité dépend alors du délai entre les potentiels d'action pré- et post-synaptiques. Plus ils sont proches l'un de l'autre et plus la modification de l'efficacité synaptique est forte. L'amplitude diminue progressivement avec l'éloignement, pour devenir nulle après quelques dizaines de millisecondes. Cette forme de plasticité est appelée *STDP* (pour *Spike-Timing-Dependent Plasticity* (Figure 1.14). Bien qu'on la présente souvent comme un mécanisme distinct de celui présenté par Hebb, on peut constater qu'elle en reprend le principe fondamental tout en le précisant sur le plan quantitatif. En ce sens, la STDP n'est qu'une variante de « synapse de Hebb ».

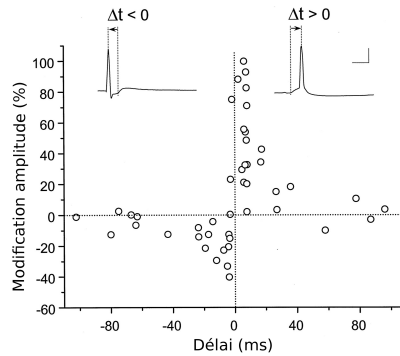


FIGURE 1.14 – Fenêtre temporelle pour l'induction de LTP et LTD. Chaque cercle représente le pourcentage de modification de l'amplitude du potentiel post-synaptique, en fonction du délai (Δt) entre le potentiel d'action du neurone post-synaptique et le potentiel d'action du neurone pré-synaptique. *Figure extraite et traduite de [Bi and Poo, 1998].*

1.3 Théories synchrones et asynchrones du fonctionnement cérébral

Le cerveau est un réseau dynamique intrinsèquement complexe. Les interactions entre les neurones de ce réseau donnent naissance aux fonctions cérébrales. Ces interactions sont caractérisées par la connectivité entre les neurones. La connectivité s'observe d'un point de vue *structurel* et d'un point de vue *fonctionnel*. La connectivité structurelle est anatomique et dépend du placement des neurones dans le réseau physique, tandis que la connectivité fonctionnelle désigne la dépendance statistique entre les activations respectives des neurones. Dans cette section, nous présentons deux théories du fonctionnement cérébral qui se focalisent sur le niveau cellulaire.

La *théorie des assemblées de neurones*, se fonde principalement sur la connexion fonctionnelle entre neurones. Les neurones appartenant à une assemblée sont identifiés par la corrélation statistique entre les temps d'occurrence des spikes qu'ils émettent, sur une durée d'observation non nulle (le spike désigne la pointe du potentiel d'actions). C'est une vision *synchrone*¹ du fonctionnement. En pratique, la mesure de la synchronisation entre les spikes tient compte d'une certaine variabilité [Grammont and Riehle, 2003].

Une autre théorie est celle des *groupes de neurones polychrones*. Elle n'est pas incompatible, mais plutôt complémentaire à celle des assemblées. Les neurones polychrones sont caractérisés par un motif d'activation *asynchrone*². Cette vision tient explicitement compte des délais de transmission entre les neurones et donc d'une forme de connectivité structurelle.

1. En informatique, le terme synchrone désigne des événements qui ont lieu en même temps. Par exemple, on considère un système à plusieurs variables comme synchrone, si toutes ses variables peuvent changer d'état à chaque temps. D'un point de vue statistique, la synchronie dépend d'une période d'observation et peut prendre en compte de la variabilité et des décalages de phase [Grammont and Riehle, 2003].

2. En informatique, le terme asynchrone désigne des événements qui n'ont pas lieu en même temps. Par exemple, on considère un système à plusieurs variables comme asynchrone si l'état d'une seule variable peut changer à chaque temps.

1.3. THÉORIES SYNCHRONES ET ASYNCHRONES DU FONCTIONNEMENT CÉRÉBRAL

Ce sont alors les temps de réception des spikes qui sont alors considérés comme synchrones sur une durée d'observation non nulle.

1.3.1 Les assemblées de neurones

Dans *The organization of behavior* [Hebb, 1949], Hebb a fourni un cadre général pour relier le comportement à l'organisation des réseaux neuronaux. L'édition originale qui date de 1949, indisponible depuis 1966, est reprise dans une nouvelle version [Hebb, 2002]. Dans cet ouvrage, Hebb a proposé l'idée de codage par *assemblées de neurones*. Il a postulé que l'activité d'une assemblée, en réponse à un *stimulus*, représenterait la perception de celui-ci. Cette théorie concerne principalement les réseaux de neurones massivement interconnectés, tels que ceux qui constituent le néocortex.

Une assemblée est un ensemble de neurones au sein duquel les connexions ont été renforcées par la plasticité synaptique. C'est dans ce contexte que Hebb a introduit la plasticité synaptique.

Lorsqu'un stimulus est appliqué sur un réseau de neurones, certains neurones sont préférentiellement activés. La stimulation répétée de ces neurones entraîne le renforcement des synapses qui les relie. Ce processus a pour conséquence d'optimiser la communication entre les neurones de l'assemblée ainsi formée. Le renforcement serait tel, que toute l'assemblée pourrait être activée même lors de la présentation partielle d'un stimulus. La mise en place des assemblées suppose donc que la répétition d'une stimulation induirait une « trace mnésique » dans le cerveau.

Les assemblées sont des structures *dynamiques*. Un neurone peut appartenir à plusieurs assemblées et peut participer à tout moment à une nouvelle assemblée traitant d'un objet ou d'une propriété similaire. Suivant le moment considéré, un neurone peut donc être impliqué dans la représentation de différents stimuli ayant des caractéristiques communes. Cette combinatoire, associée au très grand nombre de neurones qui constituent le cerveau, démultiplie les capacités de représentation des neurones, par rapport à des neurones qui seraient seulement conçus comme des unités de traitement indépendantes.

La théorie des assemblées permet de répondre au *problème de liage* (ou *binding problem* en anglais) [Von der Malsburg, 1987] [Von der Malsburg, 1995] [Von der Malsburg, 1999] (Figure 1.15). De manière schématique, supposons des assemblées qui s'activent respectivement en réponse à la couleur « bleu », la couleur « jaune », la forme « rond » et la forme « carré ». Lorsqu'un rond bleu est perçu, alors les assemblées « bleu » et « rond » sont toutes deux activées. Maintenant, si un rond bleu et un carré jaune sont perçus simultanément, les quatre assemblées s'activent. On parle alors de *superposition catastrophique* [Von der Malsburg, 1999] : l'activité simultanée de ces assemblées peut tout autant représenter un rond bleu et un carré jaune qu'un rond jaune et un carré bleu. Les différentes caractéristiques doivent donc être correctement combinées pour représenter le stimulus de manière cohérente. Un moyen de résoudre ce problème est l'hypothèse de *liage par syn-*

CHAPITRE 1. PRÉREQUIS BIOLOGIQUES

chronisation temporelle [Singer and Gray, 1995]. Selon l'hypothèse de liage temporel, les assemblées activées par les caractéristiques d'un même percept seraient liées par les temps d'émission des spikes. En reprenant l'exemple précédent, si un rond bleu et un carré jaune sont perçus simultanément, les temps des spikes émis par les neurones de l'assemblée « rond » seraient corrélés au temps de ceux émis par l'assemblée « carré ». Par ailleurs, il y aurait également une corrélation temporelle entre les neurones de l'assemblée « carré » et ceux de l'assemblée « jaune ». Ainsi, les caractéristiques des percepts « rond bleu » et « carré jaune » ne seraient pas confondues (Figure 1.15).

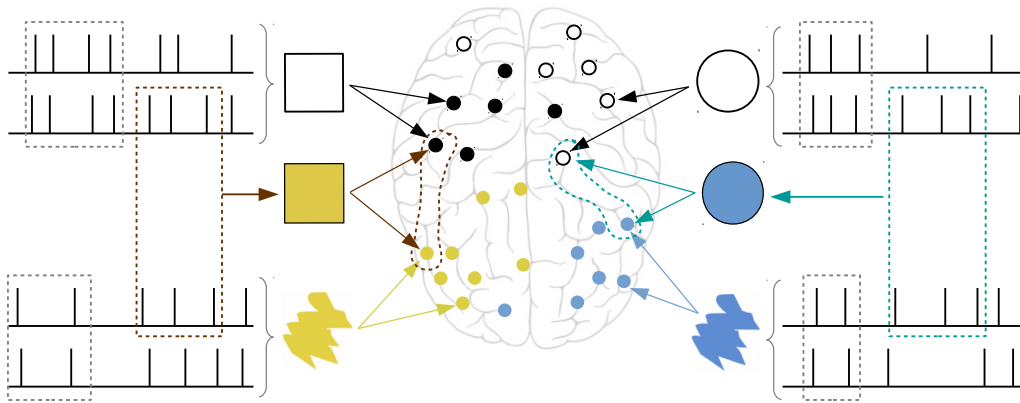


FIGURE 1.15 – Représentation schématique du problème de liage (*binding problem*). L'hypothèse de liage par synchronisation temporelle permet de regrouper les assemblées activées par des caractéristiques liées à un même percept : les temps d'émission des spikes (représentés par des barres verticales) sont alors synchronisés à l'échelle de quelques millisecondes (1 à 30). Ici, quatre assemblées sont respectivement activées par les formes « carré » ou « rond » et par les couleurs « jaune » ou « bleu ». Lors de la présentation simultanée d'un carré jaune et d'un rond bleu, les quatre assemblées sont donc activées. La cohérence de l'interprétation repose sur la synchronisation de neurones des assemblées « carré » et « jaune » en parallèle de la synchronisation de neurones des assemblées « rond » et « bleu », constituant, par définition, de nouvelles assemblées.

Finalement, c'est la corrélation des temps précis d'émission des spikes qui permettrait de définir les neurones temporairement engagés dans l'activation d'une assemblée. Cette hypothèse est appuyée par de nombreuses observations expérimentales [Gray et al., 1989] [Gray and Singer, 1989] [Engel et al., 1992] [Singer et al., 1997] [Gray, 1999] [Grammont and Riehle, 1999] [Grammont and Riehle, 2003]. En 1989, les expériences de *Charles Gray* et *Wolf Singer* ont permis de mettre en évidence la synchronisation des neurones ayant le même champ récepteur dans le cortex visuel primaire du chat, lors de la présentation d'une barre lumineuse se déplaçant dans leur direction préférentielle [Gray and Singer, 1989]. Le champ récepteur d'un neurone correspond simplement au stimulus qui déclenche son activation. Chez le singe, il a été montré que lorsque deux barres ayant des orientations différentes sont présentées, les neurones sont activés indépendamment en fonction de leurs champs récepteurs, mais aucune synchronisation n'est observée entre eux. Au contraire, lorsqu'une unique barre ayant une orientation à mi-chemin entre les deux champs récepteurs est présentée, les neurones sont activés et se synchrono-

nisent [Singer et al., 1997]. Cette observation corrobore également l'idée d'appartenance dynamique des neurones aux assemblées.

Le liage temporel est donc un moyen d'étiqueter les neurones comme appartenant à une même assemblée à un temps donné. Cette signature permettrait un traitement non ambigu des signaux et la représentation cohérente des percepts. Ceci suppose notamment que la synchronisation des neurones soit détectée et interprétée [Singer et al., 1997]. Cette supposition est longtemps restée controversée, mais elle est cependant soutenue par de multiples observations qui suggèrent que les neurones sont effectivement capables d'évaluer, avec une grande précision, les relations temporelles parmi les signaux qu'ils reçoivent en agissant comme des *détecteurs de coïncidence* [König et al., 1996] [Singer et al., 1997] [Sakurai, 1999] [Gerstner and Kistler, 2002]. Par exemple, pour localiser les sons, les neurones du noyau auditif du tronc cérébral sont capables d'évaluer le délai entre les signaux provenant des deux oreilles avec une précision de l'ordre de la microseconde [Grothe, 2003] [Köppl, 2012] [Klug and Albrecht, 2015].

La synchronisation pourrait avant tout jouer un rôle dans la création de nouvelles assemblées et dans les processus associatifs. Dans ce contexte, une assemblée de neurone est souvent caractérisée par la fameuse expression « neurons that fire together, wire together » [Lowel and Singer, 1992] qui signifie que des neurones qui s'activent ensemble se connectent entre eux.

Dans ses travaux, Hebb a également proposé l'existence de *phase-sequences* en postulant que l'activation d'une assemblée pourrait déclencher l'activation séquentielle d'autres assemblées. C'est ce mécanisme qui serait à l'origine des fonctions cérébrales complexes. L'activité d'une assemblée pourrait ainsi être la réponse à des stimuli externes (sensoriels) ou internes (autres assemblées) [Almeida-Filho et al., 2014]. Cette idée a ensuite été formalisée par *Moshe Abeles* sous le nom de *synfire chains* [Abeles, 2012].

Finalement, la théorie des assemblées de neurones est une vision *synchrone* du fonctionnement cérébral. Les neurones associés à la représentation d'un percept ou à une tâche donnée seraient définis par la corrélation temporelle de leurs activations. Cependant, à une échelle plus fine, la théorie des assemblées suppose également un mécanisme asynchrone de l'activation des neurones.

1.3.2 Les groupes de neurones polychrones

La théorie des assemblées suppose implicitement que des neurones ayant une même cible post-synaptique et s'activant de manière synchrone, entraînent l'arrivée synchrone de leurs potentiels d'action au neurone post-synaptique, favorisant l'atteinte de son seuil de dépolarisation et la transmission de l'influx nerveux. Au contraire, si des neurones s'activent de manière asynchrone, leurs potentiels d'action arrivent à des temps différents au neurone post-synaptique, induisant une faible sommation et une dépolarisation insuffisante pour déclencher l'émission d'un potentiel d'action post-synaptique. Cette vision implique que les

CHAPITRE 1. PRÉREQUIS BIOLOGIQUES

délais de transmission entre les neurones sont égaux ou négligeables.

En 2006, partant de ce constat, *Eugene M. Izhikevich* a remis en cause l'identification de groupes fonctionnels de neurones sur la base de la *synchronisation* « exacte » des temps d'émission des potentiels d'action. Il propose alors la notion de *polychronisation*, qui est une forme de coordination dans laquelle sont pris en compte les délais de transmission entre les neurones [Izhikevich, 2006]. Les *groupes de neurones polychrones* ne sont pas activés de manière synchrone mais de manière « verrouillée » les uns par rapport aux autres (Figure 1.16). La polychronie se distingue de l'asynchronie qui caractérise généralement des événements non synchrones aléatoires, n'impliquant pas ce verrouillage reproductible.

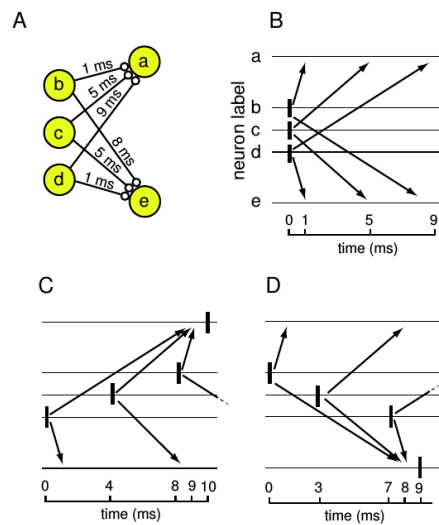


FIGURE 1.16 – Représentation schématique d'un groupe de neurones polychrones. (A) Les connexions synaptiques des neurones *b*, *c* et *d* aux neurones *a* et *e* ont différents délais de conduction axonale. (B,C,D) Les spikes émis par les neurones sont représentés par des barres verticales. Chaque flèche pointe les temps d'arrivée aux neurones post-synaptiques. (B) L'émission synchrone n'active pas les neurones post-synaptiques puisque les spikes arrivent à des temps différents. (C) Le motif d'activations constitué du neurone *d* qui s'active à 0 ms, du neurone *c* qui s'active à 4 ms et du neurone *b* qui s'active à 8 ms est optimal pour activer le neurone *a* qui reçoit les spikes de manière synchrone. (D) L'ordre inverse est optimal pour activer le neurone *e*. **Figure extraite de [Izhikevich, 2006].**

À l'inverse de la théorie des assemblées de neurones, les groupes de neurones polychrones ne sont pas caractérisés par des activations synchrones. Cependant, ces deux théories ne sont pas incompatibles. Si on considère que des neurones polychrones constituent un groupe fonctionnel, ils pourraient activer, par leurs activations asynchrones, des neurones de manière synchrones. Les neurones ainsi activés pourraient définir une assemblée (Section 1.2).

Les groupes de neurones polychrones mettent en évidence une *causalité* entre les activations, liée à la connectivité structurelle des réseaux. Izhikevich a d'ailleurs montré (par simulation) que ces groupes fonctionnels peuvent émerger, dans un réseau de neurones aléatoire, grâce à la STDP qui est effectivement une extension causale de la plasticité hebbienne.

Similairement aux assemblées de neurones, les groupes polychrones seraient dynamiques, chaque neurone pouvant être impliqué dans différents groupes, à différents moments. De plus, les neurones polychrones pourraient être impliqués dans plusieurs groupes simultanément, ce qui suggère une capacité de représentation théoriquement infinie [Izhikevich, 2006] [Martinez and Paugam-Moisy, 2008].

1.3.3 Le codage temporel : point commun des deux théories

Alors qu'un point de vue classique, mais daté historiquement, est que les neurones encodent l'information exclusivement *via* la modulation de la fréquence moyenne d'émission des potentiels d'action, la théorie des assemblées, comme celle des groupes polychrones, accorde une importance particulière aux temps précis d'émission d'occurrence des spikes, dans l'identification d'ensembles fonctionnels de neurones.

Des observations expérimentales s'accumulent effectivement en faveur d'un codage temporel de l'information neuronale, avec une précision de l'ordre de la milliseconde [Thorpe, 1990] [Stevens and Zador, 1995] [Theunissen and Miller, 1995] [Butts et al., 2007]. L'une des observations populaires en faveur du codage temporel a été énoncée par *Simon Thorpe* à la fin des années 80 [Thorpe and Imbert, 1989] [Thorpe, 1990] [Thorpe et al., 2001].

Simon Thorpe a mis en avant le fait qu'au niveau du système visuel, le traitement de l'information serait si rapide que chaque neurone impliqué émettrait un unique potentiel d'action pour transférer l'information à traiter. Des données neurophysiologiques montrent que des neurones corticaux sont capables de répondre de manière très précise à un stimulus visuel en moins d'une centaine de millisecondes. Or, entre les photorécepteurs de la rétine et le cortex temporal, il existerait au moins 10 passages synaptiques, ce qui laisserait uniquement 10 ms de traitement par neurone. Étant donné le spectre de fréquences typique de ces neurones (0 à 100 potentiels d'action par seconde), les neurones ne pourraient pas générer suffisamment de potentiels d'action pour qu'une fréquence puisse porter l'information visuelle.

Le fonctionnement cérébral est fondé sur la transmission de potentiels d'actions entre les neurones. Les potentiels d'action, étant des impulsions électriques stéréotypées, ce ne sont à l'évidence pas leurs caractéristiques qui portent l'information. Cependant, il apparaît que les temps d'occurrence des spikes, en complément des variations de fréquence permettent d'encoder l'information. La précision temporelle pourrait être portée à la fois par les motifs d'émission des neurones uniques et par les temps relatifs entre différents neurones [Izhikevich, 2006]. L'idée sous-jacente est que les temps d'occurrence des spikes déterminent la convergence des signaux pré-synaptiques et agit ainsi sur l'atteinte du seuil de dépolarisation et l'activation post-synaptique.

D'après les théories présentées dans ce chapitre, ce sont des mécanismes de plasticité qui seraient à l'origine de l'émergence des groupes fonctionnels. En particulier, la STDP permettrait de sélectionner les délais de transmission « compatibles » [Izhikevich, 2006]. L'importance de ces délais a été mise en évidence par Izhikevich. Cependant, il n'évoque que les

CHAPITRE 1. PRÉREQUIS BIOLOGIQUES

délais dus à la transmission axonale. La section précédente montre bien que les dendrites joueraient un rôle crucial dans ces groupes fonctionnels, de par leurs propriétés dont dépend largement l'intégration spatio-temporelle des potentiels d'action.

CHAPITRE 2

PRÉREQUIS MATHÉMATIQUES ET INFORMATIQUES

Dans ce chapitre, nous introduisons les concepts mathématiques et informatiques employés dans les chapitres suivants. Nous introduisons des concepts relatifs aux fonctions mathématiques, quelques notions d'algèbre et les bases de la théorie des catégories. Puis, du côté informatique, nous présentons les concepts clés relatifs aux langages synchrones et au *model checking* associé.

2.1 Fonctions mathématiques

2.1.1 Concepts élémentaires utiles

Une *fonction* $f : X \rightarrow Y$ est une relation entre un ensemble d'entrées X et un ensemble de sorties Y où chaque entrée x possède au plus une sortie $y = f(x)$. On dit alors que y est l'image de x . Si pour toute entrée il existe une image, on parle alors d'*application* sinon, l'ensemble des $x \in X$ possédant une image est appelé le *domaine* de f .

Fonctions injectives

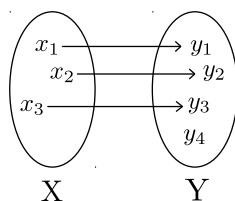


FIGURE 2.1 – Injection $X \rightarrow Y$.

CHAPITRE 2. PRÉREQUIS MATHÉMATIQUES ET INFORMATIQUES

Une fonction $f : X \rightarrow Y$ est *injective* (Figure 2.1) si et seulement si, pour tout x et $x' \in X$, $(f(x) = f(x') \Rightarrow x = x')$.

Fonctions surjectives

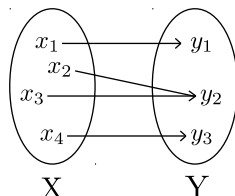


FIGURE 2.2 – Surjection $X \rightarrow Y$.

Une fonction $f : X \rightarrow Y$ est *surjective* (Figure 2.2) si et seulement si, pour tout $y \in Y$, il existe $x \in X$ tel que $f(x) = y$.

Fonctions bijectives

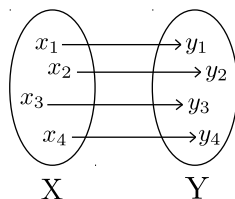


FIGURE 2.3 – Bijection $X \rightarrow Y$.

Une fonction $f : X \rightarrow Y$ est *bijjective* (Figure 2.3) si et seulement si elle possède une application inverse g telle que $f \circ g = Id_x$ et $g \circ f = Id_y$.

De manière équivalente, une application est bijective si c'est une application à la fois injective et surjective.

Fonctions continues

Intuitivement, une fonction est *continue* si sa courbe représentative peut être tracée « sans lever le crayon » sur son intervalle de définition. Plus formellement : une fonction f est continue en un point x_0 si et seulement si $\lim_{x \rightarrow x_0} f(x) = f(x_0)$, et f est continue sur un intervalle I si et seulement si elle est continue en chaque point de I .

Une autre manière d'exprimer la continuité est de dire que f est continue sur un intervalle I si et seulement si, pour tout $\varepsilon > 0$ et pour tout $x \in I$ il existe $\eta > 0$ tel que pour tout $y \in I$, $(|x - y| < \eta \Rightarrow |f(x) - f(y)| < \varepsilon)$. C'est-à-dire qu'une variation infiniment petite de $x - y$ implique une variation infiniment petite de $f(x) - f(y)$.

Il existe une notion de continuité *uniforme* qui est plus forte que la continuité simple définie ci-dessus (Figure 2.4). Dans le cas où f est uniformément continue sur I , le choix de η par rapport à ϵ est indépendant du point x considéré. L'uniformité est donc en quelque sorte, une vision « globale » de la continuité.

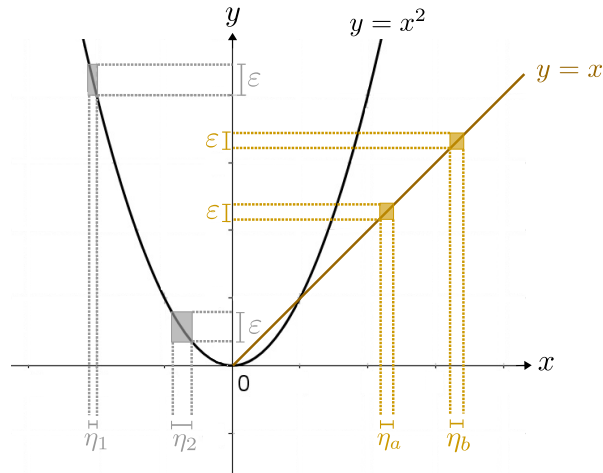


FIGURE 2.4 – La fonction identité est un exemple de fonction uniformément continue, car le choix de η en fonction de ϵ ne dépend pas de x ($\eta_a = \eta_b$). La fonction carrée est continue, mais pas uniformément continue (η tend vers 0 lorsque x tend vers $+\infty$).

Fonctions dérivables

Une fonction f est *dérivable* en un point x_0 si et seulement si $\lim_{x \rightarrow x_0} \frac{f(x) - f(x_0)}{x - x_0}$ existe et est finie. Cette limite est alors appelée la dérivée de f en x_0 (notée $f'(x_0)$) et représente la pente de la tangente à la courbe représentative de f en x_0 . De plus, f est dérivable sur un intervalle I si et seulement si elle est dérivable en chaque point de I . Toute fonction dérivable sur un intervalle est forcément continue, mais la réciproque est fautive.

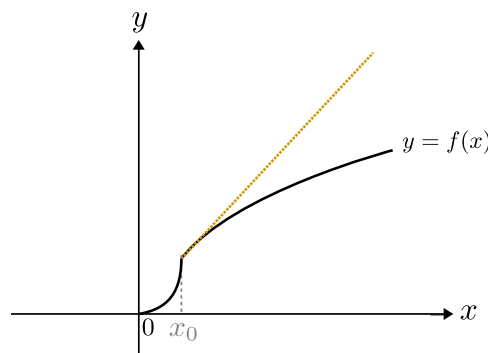


FIGURE 2.5 – La fonction f est un exemple de fonction dérivable à droite en x_0 . En jaune est représentée la demi-tangente à droite de f en x_0 . La fonction f n'est pas dérivable à gauche en x_0 et donc plus généralement f n'est pas dérivable en x_0 .

Dans certains cas, $\lim_{x \rightarrow x_0} \frac{f(x) - f(x_0)}{x - x_0}$ existe uniquement si $x > x_0$ ou si $x < x_0$. On définit ainsi

la dérivabilité à *gauche* ou à *droite* (Figure 2.5). Une fonction f est dérivable à gauche en un point x_0 si et seulement si $\lim_{x \rightarrow x_0^-} \frac{f(x) - f(x_0)}{x - x_0}$ existe et est finie. La courbe représentative de f admet alors une demi-tangente à gauche en x_0 . De même, une fonction f est dérivable à droite en un point x_0 si et seulement si $\lim_{x \rightarrow x_0^+} \frac{f(x) - f(x_0)}{x - x_0}$ existe et est finie. La courbe représentative de f admet alors une demi-tangente à droite en x_0 . Finalement, f est dérivable en x_0 si et seulement si la dérivée à gauche de f en x_0 et la dérivée à droite de f en x_0 existent et sont égales.

Fonctions lipschitziennes

Une fonction f définie sur un intervalle I est *lipschitzienne* s'il existe un réel k tel que pour tout x et y appartenant à I , $|f(x) - f(y)| \leq k|x - y|$. La fonction f est alors dite k -lipschitzienne (Figure 2.6). De plus, le plus petit réel k vérifiant cette propriété est appelé la *constante de Lipschitz* de f . Intuitivement, une fonction lipschitzienne est contrainte par k dans sa manière d'évoluer. Dans le cas particulier des fonctions dérivables, une fonction est lipschitzienne si et seulement si sa dérivée est bornée.

Contre-exemple : La fonction racine cubique dont la dérivée en 0 est infinie n'est pas lipschitzienne.

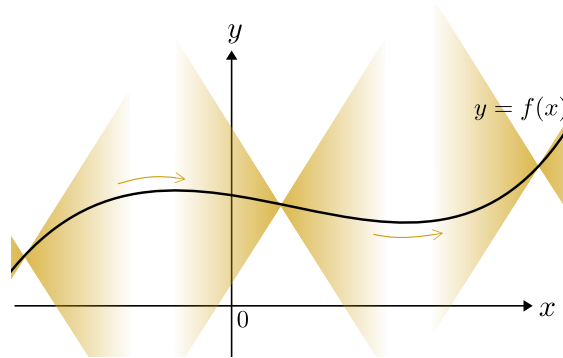


FIGURE 2.6 – La fonction f est un exemple de fonction k -lipschitzienne. Ainsi, lorsque le double cône jaune se déplace le long de la courbe représentative de f , la courbe n'en sort jamais. Plus k est petit et plus l'angle d'ouverture du cône est aigu.

Une fonction f est *localement* lipschitzienne si pour tout x appartenant à I , il existe un voisinage V de x tel que la restriction de f à V soit k -lipschitzienne. La constante k peut alors dépendre de V et donc de x .

Remarque : Toute fonction lipschitzienne est uniformément continue.

2.1.2 Les équations différentielles

Les équations différentielles sont employées très régulièrement pour modéliser des systèmes, notamment en sciences physiques ou en biologie. C'est ce que nous verrons en particulier dans les Chapitres 3 et 4.

Les *équations aux dérivées partielles* (EDPs) relient une fonction (inconnue) à plusieurs variables et à ses dérivées partielles successives. Dans cette section, on se focalisera sur le cas particulier des *équations différentielles ordinaires* (EDOs) où la fonction inconnue ne dépend que d'une seule variable et de ses dérivées successives.

Ordre des équations différentielles

Une équation différentielle du *premier ordre* relie simplement une fonction à sa dérivée première. Dans une équation différentielle du *second ordre*, la fonction est également reliée sa dérivée seconde. Plus généralement, une équation différentielle d'ordre n est une équation reliant une fonction à ses dérivées dont le degré maximal de dérivation est n . Une équation différentielle d'ordre n est donc de la forme $F(x, y, y', y'', \dots, y^{(n)}) = 0$ où F est une fonction qui dépend de $n + 2$ variables. On dit qu'une équation différentielle d'ordre n est *normale* si elle est sous la forme $y^{(n)} = f(x, y, y', y'', \dots, y^{(n-1)})$, soit sous la forme $y' = f(x, y)$ dans le cas d'une équation du premier ordre.

Exemple: $y' + 2y - x^2 = 0$ et $y''' + 2y'' + y' + (x-3)y + e^x = 0$ sont des équations différentielles respectivement d'ordre 1 et 3.

Équations différentielles linéaires

Une équation différentielle d'ordre n linéaire est de la forme $a_0(x)y + a_1(x)y' + a_2(x)y'' + \dots + a_n(x)y^{(n)} = b(x)$ où $a_0, a_1, a_2, \dots, a_n$ et b sont des fonctions numériques continues. Une équation de ce type est dite *sans second membre* (ou homogène) si $b(x) = 0$ et à *coefficients constants* si $a_0, a_1, a_2, \dots, a_n$ sont des constantes réelles.

Contre-exemple : $y' + y^2 = 3x$ et $y'' + 3y' + \cos(y) = 0$ sont des équations différentielles non linéaires.

Résolution des équations différentielles et déterminisme

Lorsque des systèmes sont représentés par des équations différentielles, l'objectif est généralement de *résoudre* ces équations. Résoudre (ou intégrer) une équation différentielle d'ordre n sur un intervalle I , c'est trouver toutes les fonctions dérivables n fois qui sont solutions de l'équation sur cet intervalle. Une fonction solution est appelée *intégrale*. Quand toutes les solutions sont de la même forme, on peut alors s'intéresser à la solution générale de l'équation.

Les systèmes gouvernés par des équations différentielles peuvent adopter des comportements très complexes, particulièrement difficiles à prédire. Néanmoins, aussi complexes qu'ils soient, ces systèmes sont par nature *déterministes*, c'est-à-dire qu'ils évoluent toujours de la même manière à partir d'une condition initiale donnée. Le caractère déterministe est une

propriété fondamentale qui se concrétise mathématiquement dans le *théorème de Cauchy-Lipschitz* (aussi connu sous le nom de Picard-Lindelöf).

On appelle *problème de Cauchy*, une équation différentielle pour laquelle on cherche une solution vérifiant une condition initiale donnée. En réponse à ce type de problème, pour une condition initiale adaptée à la forme de l'équation, et sous certaines hypothèses de régularité, le théorème de Cauchy-Lipschitz garantit l'*existence* et l'*unicité* d'une solution. Plus précisément, pour un problème de Cauchy associé à une équation différentielle de premier ordre $y'(x) = f(x, y(x))$, la condition initiale doit être de la forme $y(x_0) = y_0$. De plus, l'hypothèse de régularité requise est plus forte que la continuité : f doit être non seulement continue en x mais également *lipschitzienne* par rapport à sa seconde variable y , ce qui signifie que la constante de Lipschitz de f ne dépend pas de x .

Méthodes de résolution des équations différentielles

Un objectif « idéal » est la résolution *analytique* des équations différentielles, c'est-à-dire l'obtention de solutions explicites. Lorsque les équations sont trop complexes, leur étude peut éventuellement se focaliser sur des cas particuliers, en posant des hypothèses simplificatrices. Il existe des procédés de résolution systématique pour un certain nombre d'équations différentielles classiques, mais dans la majorité des cas, la résolution analytique s'avère impossible.

Pour pallier cette limitation, il existe des méthodes de résolution *numérique* qui permettent d'obtenir des solutions approchées, en partant d'une condition initiale. Ces techniques seront utiles dans le Chapitre 7 de cette thèse. Sans pour autant perdre en généralité, on se restreint ici aux équations différentielles du premier ordre, car les équations différentielles d'ordre supérieur peuvent toujours être réduites à un système d'équations différentielles du premier ordre en introduisant des variables supplémentaires.

Exemple: L'équation différentielle du second ordre $y''(x) = 2y(x)$ peut être réécrite sous la forme de deux équations du premier ordre : $y'(x) = z(x)$ et $z(x)' = 2y(x)$.

Les méthodes numériques s'appuient sur la discrétisation du domaine d'étude en une multitude de *pas*. L'idée est d'obtenir des expressions algébriques qui décrivent la variation de la fonction $y(x)$ lorsque la variable x est incrémentée d'un pas de taille arbitraire. De manière générale, si la taille du pas tend vers zéro, la solution approchée tend vers la solution exacte. L'implémentation littérale de cette approche est bien connue sous le nom de méthode d'*Euler*. Il s'agit de la plus simple et de la plus ancienne des méthodes numériques. Depuis Euler (1768), de nombreuses méthodes ont été développées avec pour objectif d'améliorer la qualité des solutions approchées. Parmi ces méthodes, on distingue deux grandes catégories : les méthodes à *un pas* et les méthodes à *pas multiples*, qui peuvent toutes deux être *explicites* ou *implicites* [Press et al., 1996]. La qualité des méthodes s'évalue selon trois critères principaux : la *convergence* qui atteste que l'erreur entre la solution approchée et la

solution réelle tend vers zéro avec les pas, *l'ordre* qui quantifie l'approximation et *la stabilité* qui évalue le comportement de l'erreur et donc la capacité à la contrôler.

Revenons sur la méthode d'Euler : il s'agit d'une méthode à un pas qui existe sous une version explicite et une version implicite. Soit l'équation différentielle $y'(x) = f(x, y(x))$ ayant pour condition initiale $y(x_0) = y_0$ et un pas de temps h , la méthode d'Euler *explicite* (ou *progressive*) permet de calculer directement $y(x_{n+1})$ à partir de $y(x_n)$ par approximation de la dérivée :

$$y(x_{n+1}) = y(x_n + h) \approx y(x_n) + h.y'(x_n) = y(x_n) + h.f(x_n, y(x_n)) \quad (2.1)$$

Le schéma *implicite* (ou régressif) est souvent plus stable, mais plus coûteux, car il ne permet pas ce calcul direct :

$$y(x_n) = y(x_{n+1} - h) \approx y(x_{n+1}) - h.y'(x_{n+1}) = y(x_{n+1}) - h.f(x_{n+1}, y(x_{n+1})) \quad (2.2)$$

La méthode d'Euler approxime linéairement et revient à remplacer localement, en chaque point, la courbe représentative de la fonction solution par sa tangente. Cette approximation est d'ordre 1 : si h est divisé par 2, l'erreur locale est divisée par 4 avec 2 fois plus de pas ce qui donne une erreur globale divisée par 2. Cette méthode a l'avantage d'être très simple à implémenter, mais depuis, de nombreuses améliorations ont été proposées. La méthode du *point du milieu* consiste à remplacer dans la méthode d'Euler explicite, l'approximation de la dérivée en x_n par la valeur au milieu de l'intervalle $[x_n, x_{n+1}]$. Il s'agit alors d'une méthode d'ordre 2, c'est-à-dire que l'erreur globale est de l'ordre de h^2 . Cette méthode est un cas particulier des méthodes de *Runge-Kutta*. Plus généralement, une méthode de Runge-Kutta consiste à calculer chacune des itérations en plusieurs étapes, par introduction de points intermédiaires. Plus l'ordre de la méthode est élevé et plus elle s'avère précise. La plus fréquemment utilisée est d'ordre 4 (RK4) [Brian, 2007] qui approxime l'équation différentielle $y'(x) = f(x, y(x))$ ayant pour condition initiale $y(x_0) = y_0$ et un pas de temps h de la manière suivante :

$$y(x_{n+1}) = y(x_n) + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (2.3)$$

où $k_1 = hf(x_n, y(x_n))$, $k_2 = hf(x_n + \frac{h}{2}, y(x_n) + \frac{h}{2}k_1)$, $k_3 = hf(x_n + \frac{h}{2}, y(x_n) + \frac{h}{2}k_2)$, $k_4 = hf(x_n + h, y(x_n) + hk_3)$.

Intuitivement, la dérivée obtenue est la moyenne pondérée des dérivées sur l'intervalle $[x_n, x_{n+1}]$ avec un poids plus grand pour les points du milieu. On peut noter qu'il existe également des méthodes de Runge-Kutta implicites.

Contrairement aux méthodes à un pas qui approximent la solution en x_{n+1} uniquement à partir de la valeur obtenue en x_n , les méthodes à pas multiples prennent également en compte des valeurs aux abscisses précédentes. On compte principalement parmi ces méthodes celles de *Nystrom*, d'*Adams-Bashforth-Moulton* ou de *Gear* [Goncalvès da Silva, 2007].

Finalement, quelle que soit la méthode, le pas doit être choisi minutieusement, car un pas trop grand amplifie l'erreur et peut entraîner de l'instabilité voire de la divergence. Au

contraire, un pas trop petit peut impliquer des temps de calcul importants. Le pas idéal peut varier au cours de la résolution, c'est pourquoi des méthodes ont été développées pour faire varier le pas au fur et à mesure des itérations, suivant l'erreur estimée à la précédente [Nicolas, 2004]. Nous avons présenté ici les principales méthodes de résolution numérique, mais il en existe bien d'autres que l'on ne détaillera pas [Press et al., 1996].

Par ailleurs, en parallèle des études analytiques et numériques, les équations différentielles font également l'objet d'études *qualitatives* qui se concentrent principalement sur le comportement limite et la stabilité des solutions. Ces questions sont à l'origine de la théorie des *systèmes dynamiques*.

2.1.3 Quelques principes d'induction utiles

Induction sur les entiers

L'*induction* sur les entiers est un outil fondamental de preuve. Classiquement, le principe d'induction est exprimé de la manière suivante : *Si une propriété $\psi(0)$ est satisfaite et que pour tout $n \in \mathbb{N}$, $\psi(n)$ implique $\psi(n + 1)$ alors pour tout $n \in \mathbb{N}$, $\psi(n)$ est satisfaite.* Ce principe d'induction peut également être exprimé comme suit : *Si pour tout $n \in \mathbb{N}$, pour tout $k < n$, $\psi(k)$ implique $\psi(n)$ alors pour tout $n \in \mathbb{N}$, $\psi(n)$ est satisfaite.*

Ces deux énoncés sont équivalents et peuvent être généralisés au principe d'induction *bien fondée* qui suit : *Soit un ensemble E muni d'un ordre strict $>$ et une propriété ψ sur E . Si pour tout élément minimal e , $\psi(e)$ est satisfaite et si le fait que $\psi(u)$ soit satisfaite pour tout $u < x$ implique que $\psi(x)$ soit satisfaite, alors $\psi(x)$ est satisfaite pour tout $x \in E$.*

Induction sur les réels

Le principe d'induction sur les entiers n'est pas généralisable à l'ensemble \mathbb{R}^+ car cet ensemble n'est pas *bien ordonné*, c'est à dire que tout sous-ensemble non vide de \mathbb{R}^+ n'a pas d'élément minimal. Néanmoins, \mathbb{R}^+ est *archimédien*, c'est-à-dire que pour deux nombres réels positifs inégaux, il existe toujours un multiple entier du plus petit, supérieur au plus grand. On peut donc déduire le *principe d'induction archimédienne* (via le *principe d'induction par intervalle*) suivant :

Lemme 1. [Principe d'induction par intervalle]

Soit ψ une propriété sur les nombres réels positifs. On définit le principe d'induction par intervalle comme suit :

S'il existe une suite croissante $\{t_i\}_{i \in \mathbb{N}}$ de nombres réels positifs telle que $t_0 = 0$ et $\lim_{i \rightarrow +\infty} t_i = +\infty$, et telle que $\forall i \in \mathbb{N}, (\forall t \in [0, t_i[, \psi(t)) \Rightarrow (\forall t \in [t_i, t_{i+1}[, \psi(t))$, alors $\forall t \in \mathbb{R}^+, \psi(t)$.

Preuve: Prouvons par induction sur i que $\forall i \in \mathbb{N}, \forall t \in [0, t_i[, \psi(t)$ est satisfaite.

- Si $i = 0$ alors $[0, t_0[= \emptyset$ donc $\forall t \in [0, t_0[, \psi(t)$ est satisfaite.
- Si $\forall t \in [0, t_i[, \psi(t)$ est satisfaite alors d'après le principe d'induction du lemme, $\forall t \in [t_i, t_{i+1}[, \psi(t)$ est satisfaite. Donc, considérant l'union des deux intervalles, il vient que $\forall t \in [0, t_{i+1}[, \psi(t)$ est satisfaite.

Pour tout $t \in \mathbb{R}^+$, il existe i tel que, $t_i > t$ car $\lim_{i \rightarrow +\infty} t_i = +\infty$. Par conséquent, $\psi(t)$ est satisfaite puisque $t \in [0, t_i[$. \square

Lemme 2. [Principe d'induction archimédienne]

Soit ψ une propriété sur les nombres réels positifs. On définit le principe d'induction archimédienne comme suit :

S'il existe $a \in \mathbb{R}^{+*}$ tel que $\forall t_0 \in \mathbb{R}^+, (\forall t \in [0, t_0[, \psi(t)) \Rightarrow (\forall t \in [t_0, t_0 + a[, \psi(t))$,
alors $\forall t \in \mathbb{R}^+, \psi(t)$.

Preuve: Puisque \mathbb{R}^+ est archimédien, il suffit d'appliquer le Lemme 1 avec $t_i = i \times a$. \square

2.1.4 Les segments

Pour alléger les définitions de notre cadre de modélisation (Partie 3.3), nous définissons préalablement la notion de *segment*. Un segment est une fonction qui va d'un intervalle de temps dans un ensemble. Classiquement les segments sont définis sur un intervalle de longueur finie [Muzy et al., 2017]. Nous étendons, dans la définition suivante, les segments à des intervalles infinis.

Définition 1. [Segment]

Étant donné un ensemble E , un segment à valeurs dans E est une application $\omega : [t_1, t_2] \rightarrow E$ où $(t_2 - t_1) \in \{-\infty\} \cup \mathbb{R} \cup \{+\infty\}$ est appelée la durée, avec la convention $[-\infty, +\infty] = \mathbb{R}$. Soit \star une opération binaire sur E . Si ω_1 et ω_2 sont des segments de même domaine, on note $\omega_1 \star \omega_2 : [t_1, t_2] \rightarrow E$ le segment tel que $(\omega_1 \star \omega_2)(t) = \omega_1(t) \star \omega_2(t)$ pour tout $t \in [t_1, t_2]$. Lorsque $\star = +$, on s'autorise la notation $\sum_{i=1}^n \omega_i$ pour $\omega_1 + \omega_2 + \dots + \omega_n$.

Nous allons souvent considérer des familles de segments indexés par une famille F . Si $W = \{\omega_i\}_{i \in F}$ et $W' = \{\omega'_i\}_{i \in F}$ sont des familles de segments à valeurs dans E telles que pour tout $i \in F$, le domaine de ω_i est égal au domaine de ω'_i , alors on note $W \star W' = \{\omega_i \star \omega'_i\}_{i \in F}$.

2.2 Algèbre générale

L'*algèbre* est une branche des mathématiques qui étudie :

- les *opérations algébriques* sur les nombres, constituant la branche de l'*algèbre classique* ;
- les *structures algébriques*, sous le nom d'*algèbre générale*.

Dans cette section, nous nous concentrerons sur des concepts d'algèbre générale, principalement nécessaires à la compréhension du Chapitre 6. Néanmoins, nous commençons par introduire quelques notions d'algèbre classique permettant d'aborder plus facilement les concepts suivants.

2.2.1 Quelques rappels d'algèbre classique

Une motivation majeure de l'algèbre classique est la résolution d'équations polynomiales, c'est à dire des équations de la forme $a_0 + a_1x + \dots + a_{n-1}x^{n-1} + a_nx^n = 0$ où l'*addition* (marquée par un signe +) et la *multiplication* (marquée par un signe \times) sont caractérisées par des propriétés fondamentales et des éléments particuliers, présentés dans la Table 2.1.

| Propriété | Addition | Multiplication |
|----------------------|-------------------------------------|--|
| Commutativité | $a + b = b + a$ | $a \times b = b \times a$ |
| Associativité | $(a + b) + c = a + (b + c)$ | $(a \times b) \times c = a \times (b \times c)$ |
| Distributivité* | NON | $(a + b) \times c = (a \times c) + (b \times c)$ |
| Élément neutre | 0 car $a + 0 = 0 + a = a$ | 1 car $a \times 1 = 1 \times a = a$ |
| Élément absorbant | NON | $a \times 0 = 0$ |
| Opération réciproque | Soustraction, car $(a + b) - b = a$ | Division**, car $(a \times b) \div b = a$ |

TABLE 2.1 – *Propriétés fondamentales de l'addition et de la multiplication.* * La multiplication est distributive par rapport à l'addition et à la soustraction. ** La division est l'opération réciproque de la multiplication pour les nombres différents de son élément absorbant.

Plus généralement :

- une opération est *commutative* si l'ordre des termes peut être modifié, sans changer le résultat de l'opération (ligne 1, Table 2.1) ;
- une opération est *associative* si l'ordre des calculs peut être modifié en regroupant des termes (entre parenthèses), sans changer le résultat de l'opération (ligne 2, Table 2.1) ;
- une opération est *distributive* par rapport à une autre opération si elle peut être distribuée sur les autres termes liés par la seconde opération (ligne 3, Table 2.1) ;
- l'*élément neutre* d'une opération est un nombre qui ne modifie pas son résultat (ligne 4, Table 2.1) ;
- l'*élément absorbant* d'une opération est un nombre qui est toujours le résultat de celle-ci lorsqu'il est présent (ligne 5, Table 2.1) ;
- la *réciproque* d'une opération est l'opération qui a l'effet inverse (ligne 6, Table 2.1).

De nombreuses autres propriétés, telles que les *identités remarquables* par exemple, découlent de ces propriétés fondamentales et viennent enrichir les techniques opératoires.

2.2.2 Structures algébriques

L'algèbre générale permet d'étendre les propriétés de l'algèbre classique pour les appliquer à d'autres *structures* qu'aux nombres. Une *structure algébrique* est formée d'un *ensemble*, muni d'au moins une *loi de composition* et satisfaisant des *axiomes* propres à la structure.

Une loi de composition peut être soit *interne* soit *externe*. Une loi de composition interne associe à deux éléments d'un unique ensemble, un élément du même ensemble. Une loi de composition externe associe à des éléments d'ensembles différents, un élément de l'un des ensembles.

Exemple: L'addition et la multiplication (dans l'ensemble \mathbb{N}) sont des lois de composition internes, car le résultat de l'addition ou de la multiplication de deux entiers naturels est également un entier naturel. En revanche, la multiplication d'un vecteur par un nombre réel est une loi de composition externe ayant pour résultat un vecteur (le réel est alors appelé scalaire).

Monoïde

Un *monoïde* (X, \star) est un ensemble X muni d'une loi de composition interne associative \star et d'un élément neutre x . On appelle *abélien* (ou commutatif) un monoïde dont la loi de composition interne \star est également commutative, c'est-à-dire qu'elle satisfait $a \star b = b \star a$ pour tous éléments a et b de X .

Groupe

Un *groupe* (G, \star) est un ensemble G muni d'une loi de composition interne associative \star , d'un élément neutre x et admettant un élément *symétrique* pour chaque élément de l'ensemble. Autrement dit, un groupe est un monoïde (G, \star) tel que pour tout élément a de G il existe un élément b de G tel que $a \star b = b \star a = x$ où b est appelé le symétrique de a . Le symétrique est appelé *opposé* dans le cas où \star est l'addition, et *inverse* dans le cas où \star est la multiplication. On appelle *abélien* (ou commutatif) un groupe dont la loi de composition interne \star est également commutative.

Anneau

Un *anneau* $(A, +, \times)$ est un ensemble A muni de deux lois de composition internes $+$ et \times , d'un élément neutre pour chacune des deux lois et tel que :

- $(A, +)$ est un groupe abélien ;

CHAPITRE 2. PRÉREQUIS MATHÉMATIQUES ET INFORMATIQUES

- (A, \times) est un monoïde ;
- la loi \times est distributive par rapport à $+$.

On appelle *abélien* (ou commutatif) un anneau dont la loi de composition interne \times est également commutative.

Exemple: L'ensemble \mathbb{Z} , muni des opérations d'addition et de multiplication usuelles, est un anneau commutatif.

Corps

Un *corps* $(K, +, \times)$ est un anneau commutatif pour lequel l'ensemble des éléments non nuls forme un groupe muni de la loi de composition interne \times . Tout élément non nul d'un corps a donc un inverse (en plus d'avoir un opposé de par la structure d'anneau). On appelle *abélien* (ou commutatif) un corps dont l'ensemble des éléments non nuls forme un groupe commutatif muni de la loi de composition interne \times .

Exemple: L'ensemble \mathbb{R} , muni des opérations d'addition et de multiplication usuelles, est un corps commutatif.

Espace vectoriel

Soient un corps $(K, +, \times)$ et un ensemble non vide E muni d'une loi de composition interne $+$ (appelée *somme vectorielle*) et d'une loi de composition externe \cdot (appelée *multiplication par un scalaire*).

L'ensemble E est un *espace vectoriel* sur K si :

- $(E, +)$ est un groupe abélien dont l'élément neutre est appelé *vecteur nul* ;
- la loi \cdot est distributive à gauche par rapport à la loi $+$ de E : pour tout $\lambda \in K$ et pour tous $x, y \in E$, $\lambda \cdot (x + y) = (\lambda \cdot x) + (\lambda \cdot y)$;
- la loi \cdot est distributive à droite par rapport à la loi $+$ de K : pour tous $\lambda, \mu \in K$ et pour tout $x \in E$, $(\lambda + \mu) \cdot x = (\lambda \cdot x) + (\mu \cdot x)$;
- la loi \cdot est associative par rapport à la loi \times de K : pour tous $\lambda, \mu \in K$ et pour tout $x \in E$, $(\lambda \times \mu) \cdot x = \lambda \cdot (\mu \cdot x)$;
- l'élément neutre de K pour \times (noté 1_K) est neutre à gauche pour \cdot : pour tout $x \in E$, $1_K \cdot x = x$.

On appelle les éléments de E des *vecteurs* et les éléments de K des *scalaires*.

Exemple: Pour tout $n \in \mathbb{N}^{*+}$, l'ensemble des n -uplets éléments de K forme un espace vectoriel de dimension n sur K (noté K^n). Ainsi, chaque élément de K^n est de la forme $x = (x_1, x_2, \dots, x_n)$ où les x_i sont des éléments de K .

Module

La notion de *module* généralise la définition des espaces vectoriels : pour un module, l'ensemble des scalaires forme simplement un anneau. Ainsi, soient un ensemble non vide M et un anneau $(A, +, \times)$. On note $(M, +, \cdot)$ l'ensemble M muni d'une loi de composition interne $+$ et d'une loi de composition externe \cdot . On distingue un module à *gauche*, d'un module à *droite*. L'ensemble M est un module à gauche sur A si :

- $(M, +)$ est un groupe abélien ;
- la loi \cdot est distributive à gauche par rapport à la loi $+$ de M : pour tout $a \in A$ et pour tous $x, y \in M$, $a \cdot (x + y) = (a \cdot x) + (a \cdot y)$;
- la loi \cdot est distributive à droite par rapport à la loi $+$ de A : pour tous $a, b \in A$ et pour tout $x \in M$, $(a + b) \cdot x = (a \cdot x) + (b \cdot x)$;
- la loi \cdot est associative *de gauche à droite* par rapport à la loi \times de A : pour tous $a, b \in A$ et pour tout $x \in M$, $(a \times b) \cdot x = \mathbf{a} \cdot (\mathbf{b} \cdot \mathbf{x})$;
- l'élément neutre de A pour \times (noté 1_A) est neutre à gauche pour \cdot : pour tout $x \in M$, $1_A \cdot x = x$.

Ce qui distingue un module à droite sur A d'un module à gauche est uniquement que la loi \cdot est associative *de droite à gauche* par rapport à la loi \times de A : pour tous $a, b \in A$ et pour tout $x \in M$, $(a \times b) \cdot x = \mathbf{b} \cdot (\mathbf{a} \cdot \mathbf{x})$. Si l'anneau A est commutatif, pour tous $a, b \in A$, $a \times b = b \times a$ et donc les modules à gauche et à droite sur A sont égaux. Dans ce cas, on parle alors simplement de module sur A .

Exemple: L'ensemble des vecteurs du plan dont les coordonnées sont des entiers relatifs forme un module sur \mathbb{Z} .

2.3 Théorie des catégories

La *théorie des catégories* est une branche des mathématiques créée dans les années 1940 par *Samuel Eilenberg* et *Saunders Mac Lane*, utilisée principalement dans de nombreuses branches des mathématiques et également en informatique théorique. La théorie des catégories étudie les relations entre les structures mathématiques, en faisant abstraction du contenu précis de ces structures. C'est un outil qui permet ainsi de factoriser des résultats qui seraient, sans elle, répartis dans autant de théories qu'il y a de structures.

Nous ne présentons pas ici les théorèmes classiques de la théorie des catégories de manière exhaustive, mais nous introduisons seulement les concepts nécessaires à la compréhension du Chapitre 6.

2.3.1 Catégories et diagrammes

Catégories

Une catégorie \mathcal{C} est un triplet constitué de :

- un ensemble $Obj(\mathcal{C})$, également noté $|\mathcal{C}|$, dont les éléments sont appelés *objets* de \mathcal{C} ;
- un ensemble de *morphismes* $\mu : X \rightarrow Y$, noté $Hom_{\mathcal{C}}(X, Y)$ où X et Y sont deux objets de \mathcal{C} (« Hom » vient d'*homomorphisme*, pour des raisons historiques) ;
- une loi rond $\circ : Hom_{\mathcal{C}}(Y, Z) \times Hom_{\mathcal{C}}(X, Y) \rightarrow Hom_{\mathcal{C}}(X, Z)$ où X, Y et Z sont des objets de \mathcal{C} .

Une catégorie est soumise à trois *axiomes* $C0, C1, C2$ (voir Figure 2.7) :

- **C0** : $Hom_{\mathcal{C}}(X, Y) \cap Hom_{\mathcal{C}}(X', Y') \neq \emptyset \Rightarrow X = X'$ et $Y = Y'$
Cet axiome précise qu'un morphisme ne peut avoir qu'un seul objet de départ et un seul objet d'arrivée.
- **C1** : $\exists Id_X \in Hom_{\mathcal{C}}(X, X)$ tel que :
 - $\forall f : X \rightarrow Y, f \circ Id_X = f,$
 - $\forall g : V \rightarrow X, Id_X \circ g = g.$

Cet axiome précise que la loi rond possède un élément neutre à droite et à gauche, c'est-à-dire que Id_X laisse tous les morphismes inchangés lorsqu'il est composé avec eux, que ce soit à droite ou à gauche.

- **C2** : Si $h \in Hom_{\mathcal{C}}(X_1, X_2), g \in Hom_{\mathcal{C}}(X_2, X_3), f \in Hom_{\mathcal{C}}(X_3, X_4)$, alors : $f \circ (g \circ h) = (f \circ g) \circ h.$

Cet axiome formalise l'associativité de la loi rond.

***Exemple:** Prenons l'exemple de la catégorie des ensembles Set . Les objets de Set sont les ensembles (\mathbb{R}, \mathbb{N} , etc.) et les morphismes sont les applications d'un ensemble dans un autre (si X et Y sont deux ensembles, $Hom_{Set}(X, Y)$ est la collection des applications de X dans Y). Enfin, la loi rond est la composition habituelle des applications. Cette catégorie satisfait effectivement les trois axiomes puisqu'une application a un unique ensemble de départ et un unique ensemble d'arrivée (C0). De plus, la composition possède un élément neutre à droite et à gauche (C1) et elle est associative (C2). Notons cependant que l'ensemble des ensembles n'existe pas, car le cardinal de l'ensemble des parties de E est toujours strictement supérieur à celui des parties de E (théorème de Cantor). Par conséquent la catégorie des ensembles ne devrait pas exister en tant que telle. Cependant il existe une notion d'univers (que nous ne détaillerons pas ici) qui permet de considérer la « classe » de tous les ensembles sans engendrer aucun paradoxe.*

En théorie des catégories, l'étude n'est pas centrée sur les objets, mais sur les morphismes. Les morphismes permettent de lier les objets entre eux et de les « comparer ». En fait, les morphismes conservent la structure d'intérêt des objets et il est ainsi possible de raisonner, de manière très abstraite sur un ensemble d'objets (par exemple d'ensembles) ayant des caractéristiques communes.

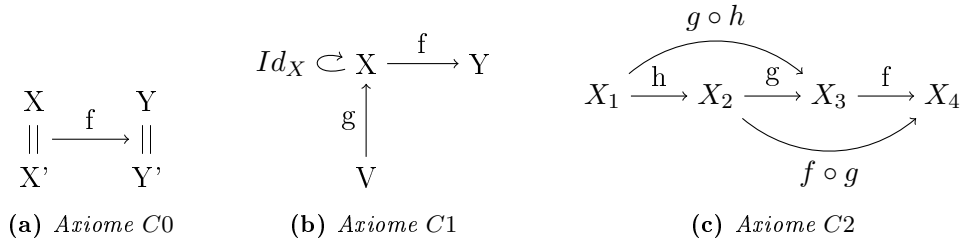


FIGURE 2.7 – Support visuel pour les axiomes des catégories.

Diagrammes

Un *diagramme* dans une catégorie \mathcal{C} est un graphe orienté tel que :

- les nœuds sont des objets de \mathcal{C} ,
- les arcs sont des morphismes $\mu : X \rightarrow Y$.

Un diagramme est *commutatif* si le résultat de la composition des morphismes sur le chemin d'un objet vers un autre est le même, quel que soit le chemin emprunté.

Dans le graphe ci-dessous (Figure 2.8), il y a deux chemins possibles pour aller de l'objet X à l'objet Y : soit directement par le morphisme f , soit par la composition des morphismes h et g , en passant par l'objet Z . Donc $f = h \circ g$ est équivalent à dire que le diagramme est commutatif.

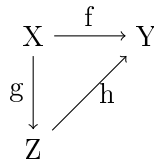


FIGURE 2.8 – Exemple de diagramme commutatif.

2.3.2 Propriétés des morphismes

En théorie des catégories, il n'est pas question « d'éléments » mais d'objets : les propriétés d'*injection*, de *surjection* ou de *bijection* de la théorie des ensembles sont donc redéfinies pour les morphismes.

Monomorphisme

Un *monomorphisme* (Figure 2.9) est un morphisme qui « simule » une injection. Un morphisme $f : X \rightarrow Y$ est un monomorphisme si, pour tout objet V et pour tous morphismes $g_1, g_2 : V \rightarrow X$, f est simplifiable à gauche avec la loi rond, c'est à dire : $f \circ g_1 = f \circ g_2 \Rightarrow g_1 = g_2$.

$$\begin{array}{ccc} & g_1 & \\ V & \xrightarrow{\quad} & X \\ & g_2 & \\ & & X \xrightarrow{f} Y \end{array}$$

FIGURE 2.9 – Monomorphisme f .

Épimorphisme

Un *épimorphisme* (Figure 2.10) est un morphisme qui « simule » une surjection. Un morphisme $f : X \rightarrow Y$ est un épimorphisme si, pour tout objet Z et pour tous morphismes $g_1, g_2 : Y \rightarrow Z$, f est simplifiable à droite avec la loi rond, c'est à dire : $g_1 \circ f = g_2 \circ f \Rightarrow g_1 = g_2$.

$$\begin{array}{ccc} & & g_1 \\ X & \xrightarrow{f} & Y \\ & & \circlearrowright \\ & & Z \\ & & g_2 \end{array}$$

FIGURE 2.10 – Épimorphisme f .

Isomorphisme

Un *isomorphisme* (Figure 2.11) est un morphisme qui « simule » une bijection. Un morphisme $f : X \rightarrow Y$ est un isomorphisme s'il existe un morphisme $g : Y \rightarrow X$ inverse de f à droite et à gauche, c'est à dire : $g \circ f = Id_X$ et $f \circ g = Id_Y$.

$$\begin{array}{ccc} & f & \\ Id_X \hookrightarrow X & \xrightarrow{\quad} & Y \twoheadrightarrow Id_Y \\ & \circlearrowleft & \\ & g & \end{array}$$

FIGURE 2.11 – Isomorphisme f .

Notons qu'un isomorphisme est à la fois un monomorphisme et épimorphisme. En revanche, il existe des morphismes qui sont des monomorphismes et des épimorphismes sans être des isomorphismes (contrairement à la théorie des ensembles). La notion d'isomorphisme est particulièrement intéressante, car elle permet de définir canoniquement, c'est à dire de manière abstraite les objets d'une catégorie « à isomorphisme près ».

2.3.3 Propriétés des objets

Étant donnée une catégorie \mathcal{C} :

- un objet I est *initial* s'il existe, pour tout objet X de \mathcal{C} , un unique morphisme de I vers X ;
- un objet T est *terminal* s'il existe, pour tout objet X de \mathcal{C} , un unique morphisme de X vers T .

Notons que le seul morphisme allant d'un objet initial ou terminal vers lui-même est donc l'identité (Figure 2.12).



FIGURE 2.12 – *Objet initial I et objet terminal T d'une catégorie.*

Quelle que soit la catégorie, deux objets initiaux I_1 et I_2 sont toujours isomorphes. En effet, l'unique morphisme de I_1 vers I_2 est l'inverse à droite et à gauche de l'unique morphisme de I_2 vers I_1 . De même, deux objets terminaux sont toujours isomorphes.

L'ensemble des morphismes d'une catégorie définit clairement une relation de *pré-ordre* (réflexive et transitive) entre ses objets.

Il n'existe pas toujours d'objet initial ou terminal dans une catégorie. De plus, un objet initial peut aussi être terminal. On peut également s'intéresser aux objets *minimaux* d'une catégorie s'il n'y a pas d'objet initial. L'ensemble des objets minimaux est un ensemble d'objets tel que :

- pour tout objet X , il existe un objet M de l'ensemble des objets minimaux et un morphisme de M vers X ,
- s'il existe un morphisme de X vers M alors X est isomorphe à M .

2.3.4 Propriétés des catégories

Catégorie opposée

Étant donnée une catégorie \mathcal{C} , la catégorie *opposée* (ou *duale*) de \mathcal{C} , notée \mathcal{C}^{op} , est la catégorie dont les objets sont ceux de \mathcal{C} avec $Hom_{\mathcal{C}^{op}}(X, Y) = Hom_{\mathcal{C}}(Y, X)$. Les catégories opposées sont parfois utiles pour la simplification de preuves, car elles « inversent » des concepts tels que les objets terminal/initial ou encore mono/épi-morphisme, mais préservent les isomorphismes.

Sous-catégorie

\mathcal{C}' est une *sous-catégorie* de \mathcal{C} si les objets et les morphismes de \mathcal{C}' sont inclus dans ceux de \mathcal{C} . De plus, \mathcal{C}' est une *sous-catégorie pleine* de \mathcal{C} si les morphismes de \mathcal{C}' entre deux objets de \mathcal{C}' sont exactement ceux de \mathcal{C} .

2.3.5 Foncteurs et transformations naturelles

Foncteurs

Les morphismes permettent de « comparer » des objets. Il est aussi possible de « comparer » des catégories grâce aux *foncteurs* qui sont des morphismes de catégories (Figure 2.13).

Un foncteur $F : \mathcal{C} \rightarrow \mathcal{C}'$ associe à tout objet X de \mathcal{C} un objet $X' = F(X)$ de \mathcal{C}' et associe à tout morphisme $\mu : X \rightarrow Y$, un morphisme $F(\mu) : F(X) \rightarrow F(Y)$. Un foncteur satisfait les propriétés suivantes :

- **F1** : $F(Id_X) = Id_{F(X)}$;
- **F2** : $F(g \circ f) = F(g) \circ F(f)$.

Ces propriétés précisent qu'un foncteur préserve l'identité de tout objet (F_1) ainsi que la composition (F_2). Les chemins entre les objets sont donc transportés par le foncteur d'une catégorie à l'autre.

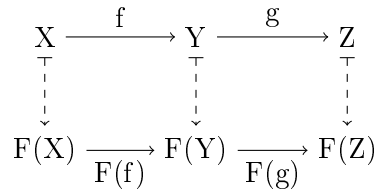


FIGURE 2.13 – Le foncteur F préserve la composition.

Un foncteur est par défaut *covariant*. Lorsqu'il est *contravariant*, un foncteur $F : \mathcal{C} \rightarrow \mathcal{C}'$ associe à tout morphisme $\mu : X \rightarrow Y$, un morphisme $\mu' : F(Y) \rightarrow F(X)$. Dans ce cas, il est parfois intéressant de raisonner sur la catégorie opposée, \mathcal{C}'^{op} car le foncteur $F : \mathcal{C} \rightarrow \mathcal{C}'^{op}$ est alors covariant.

Transformations naturelles

À un niveau d'abstraction encore supérieur, il existe également des morphismes de foncteurs, parfois appelés *transformations naturelles* (Figure 6.9). Étant donnés deux foncteurs $F, G : \mathcal{C} \rightarrow \mathcal{C}'$, une transformation naturelle de F dans G est une famille de morphisme indexée par $|\mathcal{C}|$, $\{\mu_X : F(X) \rightarrow G(X)\}_{X \in \mathcal{C}}$, telle que pour tout morphisme $\mu : X \rightarrow Y$, $\mu_Y \circ F(\mu) = G(\mu) \circ \mu_X$. En d'autres mots, il existe une transformation naturelle de F dans G si la famille $\{\mu_X\}_{X \in \mathcal{C}}$ fait commuter le diagramme de la Figure 6.9.

$$\begin{array}{ccc}
 F(X) & \xrightarrow{F(\mu)} & F(Y) \\
 \mu_X \downarrow & & \downarrow \mu_Y \\
 G(X) & \xrightarrow{G(\mu)} & G(Y)
 \end{array}$$

FIGURE 2.14 – Transformation naturelle de F dans G .

2.4 Langages synchrones et model checking

2.4.1 Les langages synchrones

Les *langages synchrones* sont des langages de programmation particulièrement adaptés pour programmer des *systèmes réactifs*. Les systèmes réactifs interagissent continuellement avec leur environnement à un rythme déterminé par cet environnement. Ils sont donc souvent *temps-réels*, c'est-à-dire sujets à des contraintes temporelles externes qui s'avèrent aussi (voire plus) importantes que l'exactitude du résultat : le système doit alors réagir correctement et suffisamment rapidement. On trouve majoritairement ce type de systèmes dans les *systèmes embarqués* dans des domaines d'applications très variés : équipements électromécaniques, dispositifs médicaux, industrie automobile, aéronautique, *etc.* Les systèmes réactifs se distinguent d'une part des systèmes *transformationnels* qui produisent un résultat à la fin de l'exécution à partir d'entrées prises au début de l'exécution et d'autre part des systèmes *interactifs* qui interagissent continuellement avec l'environnement, mais à leur propre rythme [Harel and Pnueli, 1985] [Berry, 1989].

Le paradigme synchrone s'appuie sur la notion de *temps logique* qui est une vision abstraite du temps réel. Le temps est alors une suite d'instants discrets et ordonnés. À chaque instant, tous les calculs effectués sont considérés simultanés : il n'y a pas de début et de fin d'instant ce qui synchronise les sorties avec les entrées. L'idée sous-jacente est que les calculs « ne prennent qu'un temps négligeable » par rapport à l'environnement. L'abstraction synchrone garantit ainsi un comportement *déterministe* en éliminant l'entrelacement des comportements concurrents.

Exemple: *En synchrone, si on considère deux instructions a et b , l'unique possibilité est de les exécuter simultanément, contrairement au paradigme asynchrone qui offre la possibilité d'exécuter a avant b ou b avant a , induisant du non-déterminisme si a et b agissent sur les mêmes variables. Les langages synchrones interdisent, par construction, les ambiguïtés ou les contradictions.*

Les langages synchrones sont des langages *spécialisés* qui sont généralement plus simples, plus rigoureux et plus efficaces que les langages d'usage général. Ils s'appuient sur une *sémantique formelle* qui permet de programmer de manière *sure* et facilite la *vérification*.

Les premiers langages synchrones, *Esterel* [Berry and Cosserat, 1984]

CHAPITRE 2. PRÉREQUIS MATHÉMATIQUES ET INFORMATIQUES

[Berry and Gonthier, 1992], *Lustre* [Pilaud et al., 1987] [Halbwachs et al., 1991] [Halbwachs, 1993] et *Signal* [Gautier et al., 1987] [Benveniste et al., 1991] sont français et datent des années 80. Depuis, d'autres langages de ce type ont été créés tels que *SML*, *Argos* et *SyncChart* [André, 1996] [Halbwachs, 2013].

Dans cette thèse, nous choisissons d'utiliser *Lustre*. *Lustre* est un langage simple dont le style de programmation (déclaratif) se prête bien au traitement des flots de données [André, 2005]. Il s'agit d'une structure très adaptée notamment pour modéliser les signaux échangés par les neurones.

2.4.2 Le langage Lustre

Création et utilisation

Le langage *Lustre* a été créé en 1984 par *Nicolas Halbwachs* et *Paul Caspi* dans le cadre d'un projet de recherche au laboratoire Verimag de Grenoble. Une dizaine d'années plus tard, *Lustre* a été industrialisé par la société Verilog, au cœur de l'environnement de développement *SCADE* (Safety Critical Application Development Environment). Cet outil est utilisé pour la conception de systèmes critiques, en particulier dans le domaine de l'embarqué, notamment par l'avionique (Airbus, Eurocopter).

Principes de base

L'une des spécificités de *Lustre* est d'être un langage de *flots*. Ceci signifie que toute variable est un flot de données, c'est-à-dire un couple constitué d'une horloge et d'une suite infinie de valeurs *typées*. Ainsi, une variable ne porte pas une valeur unique, mais une valeur à chaque instant logique : dire que la variable x vaut a revient donc à dire qu'à chaque instant i (où x_i est défini), x_i vaut a_i .

Lustre est un langage *déclaratif* : un programme *Lustre* est constitué d'un ensemble de déclarations définissant les relations qui existent entre les différents flots. En programmation déclarative, on décrit plutôt le problème (le *quoi*) que l'algorithme explicite pour le résoudre (le *comment*). *Lustre* privilégie une vision *fonctionnelle* de la programmation déclarative : les relations entre les flots sont alors décrites par des équations mathématiques vraies pour tous les instants. Alors que dans de nombreux langages de programmation le symbole $=$ représente l'affectation, en *Lustre* il représente l'égalité mathématique. Le compilateur *Lustre* n'autorisera donc pas la déclaration $x = x+1$: cette définition cyclique est contradictoire, car à un instant donné, la variable ne peut pas posséder à la fois une valeur et la même valeur incrémentée de 1.

Introduction au langage

En Lustre, on retrouve :

- les types de base `bool`, `int`, `real`,
- les opérateurs arithmétiques `+`, `-`, `*`, `/`, `mod`,
- les opérateurs logiques `or`, `and`, `not`,
- les opérateurs de comparaison `=`, `<>`, `<`, `>`, `>=`, `<=`.

On trouve également l'opérateur de contrôle `if then else`.

Ces opérateurs classiques sont à remettre dans le contexte des flots : il faut considérer les opérations « point à point », pour chaque instant logique.

***Exemple:** Étant donnés deux flots x et y , le flot $x + y$ vaut pour tout instant i (où x_i et y_i sont définis), $x_i + y_i$ soit la valeur de x à l'instant i additionnée à la valeur de y au même instant.*

De la même manière, l'opérateur `if then else` qui est généralement traduit par « si... alors... sinon » devrait plutôt être interprété comme « quand... alors... sinon ». Prenons l'exemple de la déclaration suivante : `x = if y then 2 else 5` où x est de type `int` et y de type `bool`. La valeur de x vaut 2 quand y est vrai et 5 sinon. La valeur du flot x est ainsi calculée en évaluant l'équation à chaque temps logique. En électronique, cet opérateur est appelé multiplexeur (Figure 2.15).

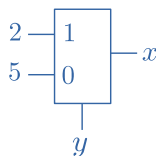


FIGURE 2.15 – Schéma logique d'un multiplexeur. Lustre permet également une représentation graphique de ce type. Cette représentation, proposée notamment dans l'environnement de développement SCADE, est particulièrement adaptée au domaine de l'électronique.

Le langage Lustre est minimaliste, mais il peut facilement être enrichi par des opérateurs construits. Pour satisfaire l'hypothèse de programmation synchrone, le langage Lustre n'autorise aucune structure qui pourrait rendre le temps de calcul non borné. En particulier, il n'y a ni boucles ni fonctions récursives. Par contre, il existe en Lustre une notion de *récursivité temporelle* qui peut être facilement employée grâce aux opérateurs temporels `pre()` et `->`. L'opérateur `pre()` permet d'accéder à la valeur d'un flot à l'instant précédent où il était défini, tandis que l'opérateur `->` (« suivi de ») concerne la valeur à tous les instants suivant le premier.

***Exemple:** La déclaration `x = 0 -> pre(x) + 1` initialise le flot x à 0 et les valeurs suivantes sont définies par un principe de récursion temporelle : aux temps suivants, x vaudra $0 + 1 = 1$ puis $1 + 1 = 2$ puis $2 + 1 = 3$ et ainsi de suite. Cette déclaration définit donc récursivement le flot x comme la suite des entiers naturels.*

CHAPITRE 2. PRÉREQUIS MATHÉMATIQUES ET INFORMATIQUES

Il existe d'autres opérateurs que l'on ne détaillera pas ici, car notre objectif n'est pas de donner une description exhaustive du langage, mais simplement d'introduire les éléments permettant d'appréhender le Chapitre 8. De plus, dans notre cas d'utilisation (application du model checking), seul un sous-ensemble du langage Lustre peut-être pris en compte, rendant certains types et opérateurs inutilisables (par exemple les tableaux).

Un programme Lustre est structuré en *nœuds*, ce qui le rend particulièrement modulaire. L'en-tête d'un nœud commence par le mot clé *node*, suivi du nom du nœud et contient la déclaration typée des flots d'entrée et des flots de sortie. Des variables locales peuvent ensuite être déclarées à la suite du mot clé *var*. Le corps du nœud, délimité par les mots clés *let* et *tel*, est un ensemble d'équations (non ordonnées) qui définissent les flots de sorties en fonction des flots d'entrée à chaque instant. Un nœud a donc la structure suivante :

```
node nomDuNoeud(Entrée1: typeE1; ...) returns(Sortie1: typeS1; ...);  
  var localVar1: typeV1; ...;  
  let  
    Équations;  
  tel
```

Le nœud principal, qui détermine les entrées et sorties du système, est choisi à la compilation.

Quelques mots sur la compilation

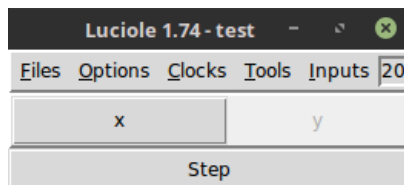
Classiquement, lorsque l'on compile un nœud Lustre, il se produit tout d'abord une *vérification statique*. Ces vérifications concernent notamment l'absence d'appels récursifs de nœuds, l'absence de déclarations non initialisées et l'absence de définitions cycliques.

La compilation permet de générer du code *séquentiel*. Il s'agit d'une boucle infinie qui représente l'écoulement du temps logique et qui contient le code correspondant à une itération du nœud principal. Ce code permet donc de lire les entrées, de calculer les sorties et de les mémoriser les valeurs utiles aux instants suivants.

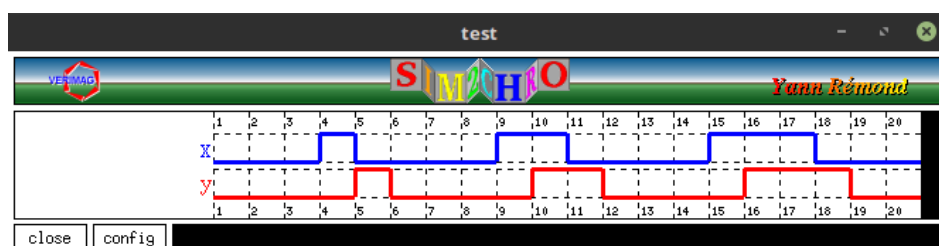
Il existe deux chemins dans la compilation. Le premier consiste à générer du code *dc* par l'outil *lus2dc*. Ce format contient un langage déclaratif qui peut également servir d'intermédiaire à d'autres langages synchrones (Esterel et Signal). À partir du code *dc*, l'outil *drac* génère du code *C*. L'autre possibilité, la plus courante, est de générer du code *ec* (expanded code) grâce à l'outil *lus2ec*. Il s'agit en fait d'un code Lustre constitué d'un seul nœud par expansion des appels aux autres nœuds. C'est ensuite l'outil *ec2c* qui se charge de produire du code *C*. Le format *ec* est le format qui sert de base aux outils de vérifications de Lustre.

Rappelons que l'ordre des équations n'a pas d'importance en Lustre. Or dans le code séquentiel généré à la compilation, l'ordre des actions correspondantes est important. Pour ordonner ces actions, un tri topologique est fait sur le graphe de dépendance des variables [Halbwachs, 1993] [Collet et al., 2001].

Une fois compilé, le code Lustre peut être exploité de différentes manières. En particulier, l'outil *Luciole* permet de simuler des nœuds Lustre, et les différents flots (entrées/sorties) peuvent être visualisés sous forme de chronogramme par *Sim2chro* (Figure 2.16).



(a) Fenêtre de simulation avec Luciole.



(b) Chronogramme avec l'outil Sim2chro.

FIGURE 2.16 – Fonctionnement de *Luciole* pour l'exemple simple d'un flot booléen y défini par $y = \text{false} \rightarrow \text{pre}(x)$ où x est également un flot booléen : y est initialisé à faux puis il prend la valeur de x à l'instant précédent. (a) Le bouton *step* de la fenêtre de simulation permet de passer à l'instant suivant et le bouton *x* permet de donner à x la valeur vraie à l'instant courant. Lorsque y est vrai, la zone grisée devient rouge. (b) Le chronogramme permet de visualiser les flots x et y .

La sous-section suivante décrit une autre possibilité en Lustre : la vérification de propriétés par *model checking*.

2.4.3 Le model checking pour Lustre

Le *model checking* est une méthode formelle qui permet de *vérifier* qu'un système satisfait une spécification sur tous les états accessibles, de manière *rigoureuse* et *automatique*. Le model checking est principalement utilisé dans la conception de circuits électroniques et la vérification de programmes, mais c'est une technique qui peut également permettre d'analyser des systèmes financiers, physiques ou biologiques [Clarke, 2008].

Le model checking nécessite une *description* adaptée du système. Généralement il s'agit d'une représentation sous forme d'*automate* explicite, c'est à dire d'un graphe orienté qui décrit les états possibles du système et son évolution. Pour vérifier qu'un programme satisfait une propriété (exprimée dans une logique appropriée, détaillée ci-dessous), le model checker vérifie de manière *exhaustive* (mais optimisée) tous les états possibles du programme, ce qui ne peut être fait par du *test*. En Lustre, chaque état correspond à une combinaison possible de valeurs pour tous les flots. La principale difficulté du model checking est donc de gérer l'*explosion* des états. Une avancée majeure dans la gestion de ce problème a été apportée les techniques *symboliques* : plutôt que de stocker individuellement chaque état, il est possible

CHAPITRE 2. PRÉREQUIS MATHÉMATIQUES ET INFORMATIQUES

de représenter des ensembles d'états de manière symbolique et compacte, notamment par diagramme de décision binaire (BDD) [Baier and Katoen, 2008] [Mebout, 2014].

Exemple: Si on considère un système à deux variables a et b et que les états possibles pour toute exécution sont (11), (01) et (10) alors, il est possible de manipuler la fonction booléenne $a \vee b$ qui représente cet ensemble (sous réserve que $a \vee b$, qui ne distingue pas les valeurs de a et b , soit pertinent pour la formule à prouver).

En plus de la description du système, le model checking a besoin d'une *logique* pour permettre d'exprimer les propriétés à vérifier. L'avantage de Lustre est qu'il bénéficie d'une définition formelle déclarative. Il est donc possible d'exprimer des propriétés dans le même langage que celui utilisé pour programmer (*via* un flot booléen). Un nœud Lustre qui encode une propriété est appelé un *observateur d'Halbwachs-Caspi* [Halbwachs et al., 1988] [Halbwachs, 1993] [Raymond, 2008]. Comme tout nœud Lustre, il s'agit d'une relation mathématique/logique entre des flots d'entrée et de sortie.

Propriétés en Lustre

En Lustre, il est seulement possible d'écrire des propriétés de *sûreté* (*safety*) [Verimag, 2018]. Intuitivement, ces propriétés assurent que « quelque chose n'arrivera jamais ». Au contraire, les propriétés de *vivacité* (*liveness*) sont des propriétés qui assurent que « quelque chose arrivera » à partir d'une certaine exécution [Lamport, 1977]. Ce type de propriétés implique de raisonner sur des exécutions infinies, ce qui nécessite généralement de faire appel à des logiques temporelles plus complexes, ou à d'autres modèles comme les réseaux de Pétri qui exploitent des techniques algébriques [Petri, 1980] [Murata, 1989].

Exemple: Pour vérifier la propriété « x ne sera jamais différent de y », on peut écrire en Lustre l'observateur (obs) suivant :

```
node obs(x, y : int) returns (prop: bool);
  let
    prop = x = y;
  tel
```

Le flot booléen `prop` est vrai quand $x = y$. La propriété est donc satisfaite si `prop` est toujours vrai.

Dans bon nombre de cas, il est possible de transformer les propriétés de vivacité en propriétés de sûreté, en utilisant la négation. De plus, pour l'étude des systèmes critiques, ce sont généralement les exigences de sûreté qui nécessitent d'être vérifiées. Dans ce contexte, la limitation de Lustre à l'écriture des propriétés de sûreté n'est donc pas trop contraignante [Verimag, 2018].

Les outils de model checking pour Lustre : le cas particulier de *Kind2*

Plusieurs outils de model checking existent pour vérifier des propriétés en Lustre. On peut notamment citer *Lesar* [Pace et al., 2004] [Verimag, 2018], *Nbac* [Jeannet, 2000] [Jeannet, 2010], *Luke* [Franzén, 2006] et *Rantanplan* [Hagen and Tinelli, 2008].

Nous allons présenter plus en détail le model checker *Kind2* [Champion et al., 2016] car c'est celui que nous utilisons dans cette thèse (Chapitre 8). Il est plus sophistiqué que les autres model checkers mentionnés ci-dessus et il a notamment été montré plus efficace pour des systèmes proches de ceux sur lesquels nous travaillons [Hagen and Tinelli, 2008] [De Maria et al., 2016]. *Kind2* est un model checker open-source qui repose sur des solveurs *SMT* (pour *Satisfiabilité Modulo Théories*). Les solveurs SMT sont des outils qui permettent de répondre au problème de la satisfiabilité d'une formule exprimée en logique du premier ordre (sans quantificateurs) combiné à des théories comme la théorie des réels, la théorie des nombres ou la formalisation de structures de données [Biere et al., 2009] [Thiré, 2016].

Exemple: Étant donnée la formule $(a > 0 \wedge a < 10) \wedge (a = b - 2) \wedge (b > 10 \vee b < 0)$, qui ne contient pas de quantificateurs (« quelque soit » \forall ou « il existe » \exists), le problème SMT consiste à déterminer si elle est satisfiable par rapport à une théorie donnée. Si on considère la théorie des réels, il s'agit de déterminer s'il existe des valeurs réelles pour a et b telles que la formule soit vraie.

Kind2 a été implémenté en se fondant sur les techniques de son prédécesseur *PKind* [Kahsai and Tinelli, 2011]. Il permet de vérifier des propriétés (de sûreté) exprimées en Lustre sous forme de *propriétés invariantes*, c'est à dire sous forme de prédicats sur les variables qui doivent être satisfaits dans tous les états atteignables. *Kind2* représente le programme Lustre par un *système de transition d'états*, c'est à dire par un ensemble d'états et de transitions entre les états, qui peuvent être étiquetées.

Pour vérifier que le système satisfait une certaine propriété, *Kind2* fait appel à plusieurs techniques qui s'exécutent simultanément et coopèrent :

- la *k-induction* qui consiste à prouver que la propriété est satisfaite dans tous les états atteignables en k transitions et que si elle est satisfaite pour k transitions consécutives alors cela implique qu'elle est satisfaite à la $k + 1^{i\text{ème}}$ transition [Gurfinkel and Ivrii, 2017] ;
- la méthode *IC3* (pour *Incremental Construction of Inductive Clauses for Indubitable Correctness*) qui génère un renforcement inductif lorsque l'induction échoue [Bradley, 2012] ; intuitivement, *IC3* permet de réexprimer une propriété inductive en s'appuyant sur une autre propriété qui est elle-même inductive et satisfaite sur les états initiaux [Sebastian, 2014] [Gurfinkel and Ivrii, 2017] ; *IC3* est souvent complémentaire avec la *k-induction* [Champion et al., 2016] [Gurfinkel and Ivrii, 2017] ;
- plusieurs moteurs qui génèrent des *propriétés invariantes auxiliaires* à la volée pour faciliter la résolution par les techniques ci-dessus [Tiwari et al., 2001] [Champion et al., 2016].

CHAPITRE 2. PRÉREQUIS MATHÉMATIQUES ET INFORMATIQUES

Kind2 est utilisé académiquement, mais également dans l'industrie, en particulier pour la conception de systèmes embarqués (Rockwell Collins, General Electric, NASA Ames) [Champion et al., 2016]. Dans le Chapitre 8 de cette thèse, nous présentons une utilisation de *Kind2* pour la vérification de propriétés sur des modèles de neurones.

CHAPITRE 3

ÉTAT DE L'ART ET MOTIVATIONS

Nos travaux ne s'inscrivent pas de manière évidente dans un courant qui possède une bibliographie établie. Par conséquent, l'état de l'art dans lequel se positionne cette thèse est relativement hétérogène. Nous choisissons donc de diviser ce chapitre en plusieurs sections bien distinctes.

Puisque notre objectif principal est de définir un nouveau modèle de neurone unique, nous commençons par introduire les modèles existants (Section 3.1). Le but ici n'est pas d'en donner une liste exhaustive, mais simplement de fournir les informations nécessaires pour mettre en évidence notre positionnement original par rapport à l'état de l'art de ce domaine.

Notre objectif à plus long terme est d'appliquer de manière systématique des méthodes formelles de l'informatique pour permettre de raisonner, de manière assistée par ordinateur, sur des neurones en réseaux. Dans une seconde section, nous présentons donc des exemples de formalismes pour modéliser des systèmes biologiques dynamiques, qui bénéficient largement des méthodes formelles et en particulier du *model checking* (Chapitre 2, Section 2.4). Nous pourrions constater, à cette occasion, que le succès des méthodes formelles repose sur un niveau d'abstraction suffisant du système à étudier (Section 3.2).

Enfin, nous expliciterons les motivations amenant à la définition du modèle de neurone unique proposé dans cette thèse (Chapitre 4). L'idée est la suivante : nous souhaitons un modèle suffisamment abstrait pour tirer profit, à terme, des méthodes de preuve formelle de l'informatique, tout en étant biologiquement plausible dans le contexte de l'étude des assemblées de neurones (Section 3.3).

3.1 Modèles de neurones uniques

Il n'existe pas de catégorisation universelle des modèles de neurones uniques, mais nous nous efforçons dans cette section de différencier les modèles à la fois en fonction de leur domaine d'application et en fonction des paradigmes et hypothèses sur lesquels ils reposent. Il n'y a pas de frontière stricte entre les différentes catégories, mais plutôt des sortes d'ordres partiels entre les modèles, selon le point de vue.

3.1.1 Modèles bio-inspirés versus modèles biologiquement plausibles

Le neurone est un système complexe dont la modélisation peut se faire à différents niveaux d'abstraction. Le niveau d'abstraction choisi dépend de multiples facteurs, notamment de l'échelle à laquelle on se place, et surtout de l'objectif de la modélisation. On distingue souvent deux types de modèles en fonction des objectifs d'étude : les *modèles biologiquement plausibles* et les *modèles bio-inspirés*. De manière caricaturale : les modèles biologiquement plausibles ont pour objectif de représenter le plus fidèlement possible les caractéristiques du neurone. Les modèles bio-inspirés, quant à eux, favorisent généralement un niveau d'abstraction élevé, souvent plus adapté à des objectifs informatiques tels que la capacité à assister les raisonnements par ordinateur ou l'efficacité algorithmique. L'intersection entre ces deux catégories est non vide : il est tout à fait possible qu'un modèle efficace pour l'informatique soit biologiquement plausible. De plus, la notion de « biologiquement plausible » est dépendante des connaissances biologiques du moment ; par exemple, le modèle de *McCulloch et Pitts* (décrit dans la section suivante) était qualifié de biologiquement plausible à l'époque de sa définition, alors qu'il est actuellement considéré comme bio-inspiré. La situation est similaire pour la notion de « bio-inspiré » : elle dépend à la fois des connaissances dans le domaine de l'informatique, des capacités de calcul disponibles et de l'objectif final du modèle.

Modèles biologiquement plausibles

Souvent, les biologistes considèrent un modèle comme biologiquement plausible s'il est très détaillé, tout en représentant au mieux le comportement des neurones biologiques. Plus généralement, on attend de ces modèles d'être capables de reproduire des observations expérimentales avec précision et d'être dotés de bonnes capacités de prédiction sous de nombreux points de vue.

L'exemple le plus caractéristique de cette catégorie de modèles de neurone est le modèle d'*Hodgkin-Huxley* [Hodgkin and Huxley, 1952d] qui est largement utilisé en neurosciences computationnelles. Il est notamment en première place de la comparaison d'Izhikevich [Izhikevich, 2004] du point de vue de la plausibilité biologique. En 1952, *Alan Lloyd Hodgkin* et *Andrew Huxley* ont publié une série de cinq ar-

3.1. MODÈLES DE NEURONES UNIQUES

tibles décrivant les mécanismes ioniques à l'origine de la génération du potentiel d'action [Hodgkin et al., 1952] [Hodgkin and Huxley, 1952a] [Hodgkin and Huxley, 1952b] [Hodgkin and Huxley, 1952c] [Hodgkin and Huxley, 1952d]. Ils menèrent des travaux expérimentaux principalement sur l'axone géant de calmar qui a l'avantage d'être suffisamment large (jusqu'à 1mm de diamètre) pour être manipulé par les techniques disponibles à cette époque [Young, 1936]. C'est à partir de ces expériences qu'ils ont défini un modèle mathématique permettant de décrire les fondements du fonctionnement neuronal, en modélisant la membrane du neurone par un circuit électrique. Le Prix Nobel de Physiologie ou Médecine leur a été attribué en 1963 [Schwiening, 2012].

Le modèle d'Hodgin-Huxley est un ensemble d'équations différentielles non linéaires (Équations 3.1 à 3.4 plus loin). Ces équations permettent de décrire l'évolution du potentiel de membrane (U) en fonction du courant qui est constitué d'une composante capacitive (I_c) et d'une composante ionique (I_i) [Hodgkin et al., 1952] (Figure 3.1) :

$$I = I_c + I_i = C \frac{dU}{dt} + I_i$$

Pour mieux comprendre cette dernière équation, rappelons que la membrane isole l'intérieur du neurone du milieu extracellulaire (Chapitre 1, Section 1.1). Elle agit ainsi comme un condensateur. Lorsqu'un courant est injecté dans la cellule (*via* la synapse ou expérimentalement), une fraction du courant charge le condensateur par accumulation de charges à la membrane tandis que des charges passent par les canaux trans-membranaires. Chaque type de canal est représenté par une résistance propre, qui peut être constante (pour les canaux qui sont supposés toujours ouverts) ou variable (pour les canaux dont l'état varie). Le modèle Hodgkin-Huxley considère de manière explicite uniquement les canaux sodiques et potassiques. Le courant ionique peut donc être lui-même divisé en un courant porté par des ions potassiques (I_K), un courant porté par des ions sodiques (I_{Na}) et un courant de fuite porté par d'autres types d'ions qui ne sont pas décrits de manière explicite (I_l) [Hodgkin and Huxley, 1952a] :

$$I_i = I_{Na} + I_K + I_l$$

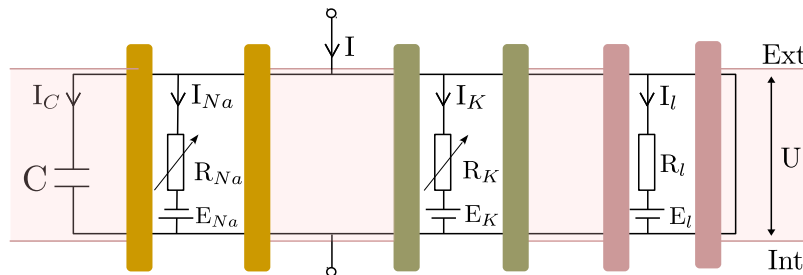


FIGURE 3.1 – Représentation de la membrane neuronale par un circuit électrique. La membrane (en rose) se comporte comme un condensateur (C) en isolant l'intérieur du neurone (Int) du milieu extracellulaire (Ext). Les canaux ioniques ont l'effet d'une résistance (R) au travers desquels peuvent se déplacer des ions.

Chacun des courants peut être décrit en termes de conductances ioniques spécifiques (g_{Na}, g_K, \bar{g}_l), chaque conductance étant égale à l'inverse de la résistance propre au type de canal [Hodgkin and Huxley, 1952d] :

$$I_{Na} = g_{Na}(U - U_{Na})$$

$$I_K = g_K(U - U_K)$$

$$I_l = \bar{g}_l(U - U_l)$$

où $U = E - E_r$, $U_{Na} = E_{Na} - E_r$, $U_K = E_K - E_r$ et $U_l = E_l - E_r$ avec E_{Na} et E_K , qui agissent comme des piles, sont respectivement le potentiel d'équilibre des ions sodiques et des ions potassiques (E_l est le potentiel pour lequel le courant de fuite est nul).

Finalement, les différentes conductances (sauf la conductance de fuite qui est constante) sont exprimées en fonction du temps et de la valeur courante du potentiel ce qui mène à l'équation différentielle non linéaire suivante :

$$I = C \frac{dU}{dt} + \bar{g}_K n^4 (U - U_K) + \bar{g}_{Na} m^3 h (U - U_{Na}) + \bar{g}_l (U - U_l) \quad (3.1)$$

avec

$$\frac{dn}{dt} = \alpha_n(U)(1 - n) - \beta_n(U)n \quad (3.2)$$

$$\frac{dm}{dt} = \alpha_m(U)(1 - m) - \beta_m(U)m \quad (3.3)$$

$$\frac{dh}{dt} = \alpha_h(U)(1 - h) - \beta_h(U)h \quad (3.4)$$

où \bar{g} représente la valeur de conductance maximale, n , m et h sont des grandeurs sans dimension associées à l'état d'activation ou d'inactivation des canaux.

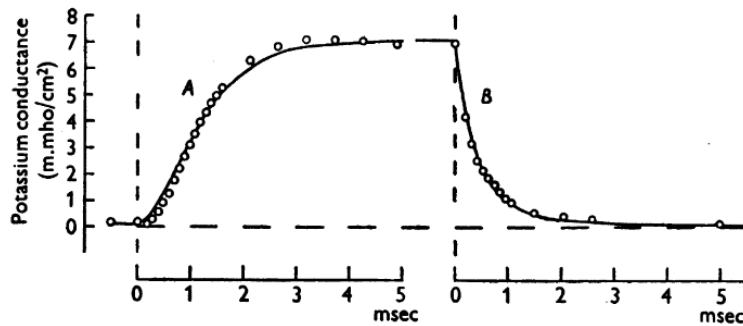


FIGURE 3.2 – *Reproduction des données expérimentales par le modèle Hodgkin-Huxley. (A) Augmentation de la conductance au potassium associée à une dépolarisation de 25 mV. (B) Diminution de la conductance associée à la repolarisation. Les cercles représentent les mesures expérimentales et la courbe pleine représente l'équation de Hodgkin-Huxley avec les paramètres particuliers. Figure extraite de [Hodgkin and Huxley, 1952d].*

Le modèle Hodgkin-Huxley a rencontré un incroyable succès depuis sa définition, autant dans la description (Figure 3.2), que dans la prédiction de propriétés [Purves et al., 2001a]. De

3.1. MODÈLES DE NEURONES UNIQUES

plus, c'est un formalisme puissant puisqu'il permet d'incorporer d'autres types de canaux ioniques [Gerstner et al., 2014]. En revanche, c'est un modèle complexe qui rend difficile l'analyse, même numérique, en particulier lorsqu'il s'agit de modéliser un grand nombre de neurones [Catterall et al., 2012]. De plus, le grand nombre de paramètres rend l'interprétation et la validation des résultats délicate [Popper, 2014].

De nombreux modèles ont permis de simplifier le modèle *Hodgkin-Huxley* tels que les modèles de *FitzHugh-Nagumo* [FitzHugh, 1961], de *Hindmarsh-Rose* [Hindmarsh and Rose, 1984] [Barrio et al., 2017] ou de *Morris-Lecar* [Morris and Lecar, 1981]. Ce sont des modèles, sous la forme de systèmes de deux (ou trois) d'équations différentielles non linéaires, qui restent tout de même très complexes.

Les modèles biologiquement plausibles contemporains sont donc des modèles très détaillés, qui se focalisent sur les mécanismes biophysiques moléculaires. Ils ont l'avantage principal d'être fidèles au comportement des neurones biologiques, mais l'inconvénient non négligeable d'être compliqués. Le niveau de détail de ces modèles rend difficile le passage à l'échelle, impliquant d'importantes ressources de calcul. Prenons l'exemple du *Blue Brain Project* [EPFL, 2018]. Il s'agit d'un ambitieux projet, créé en 2005 à l'*École Polytechnique Fédérale de Lausanne* (EPFL, Suisse), dirigé par *Henry Markram* et doté d'un budget d'1,2 milliard d'euros sur 10 ans, principalement financé par l'Union européenne. Il a pour objectif de modéliser et de simuler de manière détaillée un cerveau de rongeur et à terme un cerveau humain. La modélisation se sert de données expérimentales à différentes échelles qui, bien que partielles, aident à contraindre le système. Grâce à la simulation, il est possible de suivre l'activité électrique des neurones au cours du temps, offrant des possibilités potentiellement illimitées d'expériences *in silico*. Cependant, la simulation d'un tel système complexe nécessite une puissance de calcul considérable, uniquement possible *via* des supercalculateurs. Le Blue Brain Project bénéficie du supercalculateur *BlueGene/Q* d'IBM, l'une des architectures les plus performantes au monde, capable d'effectuer des millions de milliards d'opérations à la seconde (20 pétaflops) [Kurzweil, 2011] [IBM, 2012] [TOP500, 2018]. Ce supercalculateur est théoriquement capable de reproduire un cerveau de rat composé de 200 millions de neurones. Il n'est clairement pas suffisant pour envisager la modélisation d'un cerveau humain qui contient près de 100 milliards de neurones. Notamment pour cette raison et plus généralement pour son irréalisme, le projet est vivement critiqué [le journal international, 2013] [Kupferschmidt, 2015] [technologies clés 2020, 2018]. En 2015, un article, publié dans *Cell* par plus de 50 chercheurs impliqués dans le Blue Brain Project, annonce la reconstruction informatique et la simulation d'un ensemble d'environ 30 000 neurones connectés par près de 400 millions de synapses. C'est une prouesse technique au vu du niveau de détails adopté, mais qui reste dérisoire par rapport aux objectifs affichés : la simulation ne concerne en effet que l'équivalent de moins d' 1mm^3 de néocortex de souris [Markram et al., 2015].

En plus du délicat passage à l'échelle, les modèles biologiquement plausibles font face à des difficultés quant à la portée de leur simulation. Effectivement, ces modèles reproduisent avec un haut degré de précision les mesures expérimentales, mais de par leur grande complexité, ils

restent difficiles à analyser, et surtout, apportent peu d'explications causales des phénomènes observés [Eliasmith et al., 2012] [Gerstner et al., 2014].

Modèles bio-inspirés

Les modèles bio-inspirés sont des modèles relativement simples qui visent à abstraire les neurones biologiques pour ne représenter que leurs propriétés fondamentales, pour le point de vue adopté. Ce type de modèles se concentre sur les propriétés algébriques, logiques, analytiques ou calculatoires du fonctionnement neuronal sans détailler les mécanismes moléculaires qui en sont à l'origine. L'approche est généralement plus phénoménologique, fondée sur des approximations des fonctions d'intérêt.

Actuellement, l'exemple le plus représentatif de modèle bio-inspiré est le *neurone de McCulloch et Pitts* ou *neurone formel* qui fut introduit par *Warren McCulloch* et *Walter Pitts* en 1943 [McCulloch and Pitts, 1943]. Ce modèle possède des entrées qui représentent les signaux reçus aux synapses et calcule une sortie qui correspond au signal à transmettre à d'autres neurones *via* l'axone. Il s'agit d'un neurone *binnaire* c'est-à-dire que les entrées et la sortie portent uniquement les valeurs 0 ou 1. Néanmoins, un coefficient numérique est associé à chaque entrée pour représenter le poids et le type de la synapse : un poids positif symbolise une synapse excitatrice alors qu'un poids négatif symbolise une synapse inhibitrice. Pour calculer sa sortie, le neurone effectue simplement la somme de ses entrées, pondérée par les poids synaptiques, à laquelle il applique une *fonction d'activation* à seuil.

Dans la définition d'origine, le neurone de McCulloch et Pitts peut-être mathématiquement défini par la formule suivante (Figure 3.3) :

$$O = f \left(\sum_{j=0}^n w_j I_j - \vartheta \right) \quad (3.5)$$

où O est la sortie, f est la fonction de *Heaviside* (ou fonction *step*), qui vaut 0 pour les réels strictement négatifs et 1 pour les réels positifs. L'ensemble I_1, \dots, I_n constitue les entrées du neurone et w_1, \dots, w_n les poids synaptiques associés aux entrées. La variable ϑ est le seuil d'activation du neurone, auquel est comparée la somme pondérée des entrées. La sortie du neurone vaut donc 0 si la somme est inférieure au seuil, et 1 lorsque le seuil est atteint. La sortie, lorsqu'elle vaut 1, représente en fait l'émission d'un potentiel d'action par le neurone.

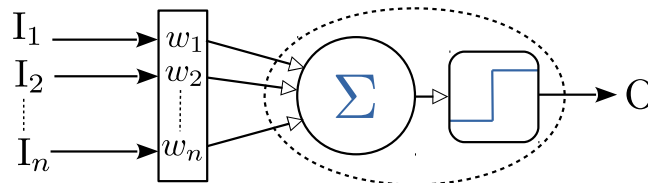


FIGURE 3.3 – Représentation schématique du neurone de McCulloch et Pitts.

Le neurone de McCulloch et Pitts est un modèle extrêmement simplifié, et basé sur l'hypothèse que les capacités complexes du cerveau pourraient émerger d'un assemblage pertinent

de neurones effectuant chacun des calculs très simples, tels que des opérations logiques basiques. Le neurone de McCulloch et Pitts est en effet capable de réaliser des calculs logiques élémentaires tels que les *ET* et *OU logiques* (Figure 3.4). Un neurone de ce type seul ne peut cependant pas calculer le *OU exclusif*. En revanche, en les combinant en réseau, ils peuvent alors calculer des fonctions arbitrairement complexes [Leshno et al., 1993]. En effet, McCulloch et Pitts ont montré qu'un réseau de neurones formels a la puissance de calcul d'une machine de Turing.

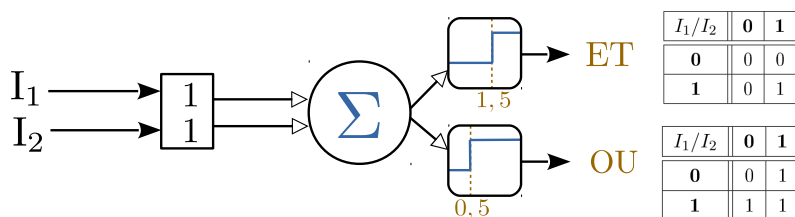


FIGURE 3.4 – Calcul des fonctions logiques *ET* et *OU* par un neurone formel à deux entrées I_1 et I_2 ayant des poids synaptiques valant 1. Les tables de vérité de cette figure présentent la valeur en sortie du neurone en fonction des entrées. La fonction calculée dépend de la valeur du seuil : par exemple, le neurone peut calculer un *ET* avec un seuil à 1,5 alors qu'il calcule un *OU* si le seuil est égal à 0,5.

Le neurone de McCulloch et Pitts est du type « tout ou rien », c'est-à-dire que sa sortie vaut 0 ou 1, mais il existe des variantes pour lesquels la fonction d'activation (f dans l'équation 3.5) est autre que la fonction de Heaviside. La sortie peut alors être réelle et non plus binaire. Parmi les fonctions d'activation les plus utilisées on trouve les fonctions identité, sigmoïdes, tangentes hyperboliques, softmax ou les rectifieurs (ReLU), mais il en existe bien d'autres [Cybenko, 1989] [Bishop, 1995] [Haykin, 2004] [Priddy and Keller, 2005] [Matignon, 2005] [Nair and Hinton, 2010] [Wikipédia, 2018].

En regard des connaissances biologiques actuelles, le modèle de McCulloch et Pitts est une approximation des fonctions du neurone, qui ne sert plus pour la compréhension du fonctionnement neuronal. Bien que ce modèle de neurone ne soit pas biologiquement plausible, il formalise de manière simple la question fondamentale « comment calcule le neurone ? », ce qui lui vaut d'être l'unité de base des réseaux de neurones artificiels actuellement utilisés dans de nombreuses applications d'intelligence artificielle.

À la fin des années 50, *Frank Rosenblatt* introduit le *perceptron* [Rosenblatt, 1958], notamment en se basant sur les travaux de Hebb [Hebb, 2002] (Chapitre 1). Il s'agit d'un algorithme faisant évoluer les poids synaptiques, permettant à un neurone formel d'apprendre. Dans sa première version, le perceptron était monocouche puis le passage au perceptron multicouches (*MLP* pour *Multi-Layer Perceptron*) a permis d'aborder des problèmes non linéaires [Minsky and Papert, 1969]. Depuis, les réseaux d'apprentissage tendent à adopter des architectures profondes, avec de plus en plus de couches cachées. Cette technique, appelée *deep learning*, permet d'atteindre des performances impressionnantes [Bengio et al., 2009]. En parallèle, de nouveaux algorithmes d'apprentissage ont été développés et notamment l'algorithme de *rétropropagation du gradient* qui est très largement exploité actuellement

en raison de son efficacité algorithmique [Kaiser and Angell, 1957] [Rumelhart et al., 1985] [Widrow and Lehr, 1990] [Touzet, 1992] [Nielsen, 2015].

Le développement des architectures profondes est notamment inspiré du cerveau qui possède lui-même une architecture profonde [Yoshua, 2018]. L'application des algorithmes d'apprentissage sur de telles architectures, pouvant atteindre des milliards de paramètres, est rendue possible par le haut niveau d'abstraction des neurones artificiels [Denil et al., 2013] [Shazeer et al., 2017]. Aucun de ces modèles ne prend en compte l'intégration dendritique.

3.1.2 Modèles fréquentiels versus modèles impulsionnels

Les modèles de neurones peuvent également être catégorisés suivant les paradigmes sur lesquels ils reposent et notamment, suivant le *codage neuronal* qu'ils produisent. Le codage neuronal, qui fait l'objet de nombreux débats, est la manière dont les potentiels d'action représentent l'information et plus généralement la manière dont se fait le traitement de l'information dans l'ensemble du cerveau.

Plusieurs types de codage ont été identifiés au vu des régularités observées sur les réponses des neurones biologiques. Deux grandes théories s'opposent : le *codage par fréquence* qui suppose que l'information est encodée dans la fréquence des potentiels d'action et le *codage temporel* qui suppose que l'information est encodée dans les temps précis d'occurrence des potentiels d'action, et dans la coordination de ces temps d'occurrence entre les neurones (Chapitre 1, Section 1.3). De ce fait, on distingue les modèles de neurones *fréquentiels* qui représentent un codage par fréquence de décharge et les modèles *impulsionnels* qui focalisent sur les temps d'occurrence des potentiels d'action.

Modèles fréquentiels

Traditionnellement, on a longtemps considéré que seule la fréquence de décharge encodait l'information neuronale [Adrian, 1928] [Hubel and Wiesel, 1962] [Kandel et al., 2012]. Ainsi, plus le stimulus est fort et plus la fréquence des potentiels d'action en réponse à ce stimulus est importante. La fréquence est une moyenne temporelle exprimée en Hertz, généralement observée sur une fenêtre temporelle de 10 à 100 ms [Guyonneau, 2006]. Une diminution de la fenêtre compensée par une répétition des essais permet d'obtenir une fréquence pouvant être assimilée à une valeur instantanée qui représente finalement une probabilité de décharge au cours du temps.

L'exemple le plus simple de modèle fréquentiel est le neurone formel. Sa sortie, surtout lorsqu'elle n'est pas binaire, peut effectivement être interprétée comme une fréquence de décharge instantanée. Ce type de modèle est bien connu d'un point de vue théorique et largement utilisé, en particulier pour l'intelligence artificielle [Bishop, 1995].

Modèles impulsionnels

Les modèles fréquentiels représentent un codage de l'information relativement simple, mais ils ne permettent pas d'étudier des phénomènes temporels fins. En particulier, ils ignorent les phénomènes de synchronisation observés expérimentalement [Meister et al., 1995] [Usrey and Reid, 1999] [Singer, 1999]. Les modèles impulsionnels se focalisent, eux, sur le temps précis d'occurrence des potentiels d'action. Ces modèles sont également appelés *spikants*, en référence à la pointe des potentiels d'actions qui porte le nom de *spike*.

Le modèle typique de neurone impulsionnel, qui est aussi utilisé en fréquentiel, est l'Integrate-and-Fire qui fut introduit par *Louis Lapicque* dès 1907 [Lapicque, 1907] [Abbott, 1999] [Brunel and Van Rossum, 2007]. De manière similaire au modèle Hodgkin-Huxley, ce modèle se fonde sur la représentation de la membrane sous forme d'un circuit électrique, mais la représentation est largement simplifiée, car l'Integrate-and-Fire est un modèle beaucoup moins détaillé au niveau moléculaire (Figure 3.5) [Eccles, 1958] [Tuckwell, 1988].

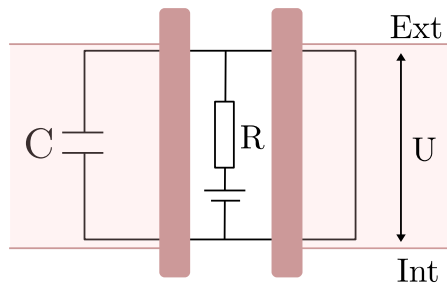


FIGURE 3.5 – Représentation de la membrane neuronale dans le modèle *Leaky Integrate-and-Fire*. La membrane (en rose) se comporte comme un condensateur (C) en isolant l'intérieur du neurone (Int) du milieu extracellulaire (Ext). Les canaux ioniques ont l'effet d'une résistance (R) au travers desquels peuvent se déplacer des ions.

La variable d'intérêt de ce modèle est le potentiel de membrane et les signaux reçus par le neurone sont représentés par un courant (qui représente le courant généré à la synapse ou une injection expérimentale). Dans sa forme la plus simple, le potentiel intègre littéralement le courant et évolue selon une équation différentielle linéaire du premier ordre :

$$C \frac{dU}{dt} = I(t) \quad (3.6)$$

où U est le potentiel, I est l'intensité du courant et C est la capacité. Cette équation décrit l'évolution du potentiel en fonction du temps, uniquement en régime sous-liminaire c'est à dire lorsque U est inférieur au seuil. En effet, une condition de seuil est ajoutée au modèle : lorsque le potentiel atteint un certain seuil, un potentiel d'action (ou spike) est émis par le neurone. Lors de l'émission d'un spike, le potentiel est réinitialisé à sa valeur de repos. De plus, à ce moment même, le seuil prend temporairement une valeur infinie pour représenter la période réfractaire absolue, durant laquelle le neurone ne peut émettre de spike (Chapitre 1, Section 1.1).

Il est important de noter que le déroulement des potentiels d'action n'est pas représenté explicitement dans ce modèle. La sortie du neurone est donc entièrement décrite par les temps d'occurrence de la pointe des potentiels d'action émis. Les modèles de type Integrate-and-Fire supposent ainsi que les potentiels d'action sont des événements ponctuels stéréotypés, ce qui est une hypothèse biologique courante (Chapitre 1, Section 1.1).

Une version plus élaborée d'Integrate-and-Fire considère également l'effet de la résistance portée par les canaux de fuite. Ce modèle, appelé *Leaky Integrate-and-Fire*, est gouverné par l'équation suivante :

$$\tau \frac{dU}{dt} = -U(t) + RI(t) \quad (3.7)$$

où R est la résistance membranaire, $\tau = RC$ et U est la valeur du potentiel par rapport à sa valeur de repos. Grâce au caractère linéaire du modèle Leaky Integrate-and-Fire, il est possible de résoudre de manière exacte sa dynamique sous-liminaire [Tuckwell, 1988]. En revanche, le mécanisme de réinitialisation rend ce système dynamique complexe.

Il existe également des modèles de type Integrate-and-Fire *non linéaires* dont la forme générale est la suivante :

$$\tau \frac{dU}{dt} = f(U) + R(U)I(t)$$

En comparaison avec le classique Leaky Integrate-and-Fire, la fonction $R(U)$ peut être interprétée comme une résistance dépendante de la valeur du potentiel et $f(U)$ rend la fuite non plus directement dépendante de U , mais d'une fonction de U . La dynamique de ces modèles est ainsi plus fidèle à la biologie [Izhikevich, 2004] [Badel et al., 2008] [Touboul, 2009]. On peut citer parmi les modèles *non linéaires*, l'Integrate-and-Fire quadratique [Mishra et al., 2006] [Gerstner et al., 2014] pour lequel $f(U) = U^2$ et le modèle exponentiel [Fourcaud-Trocmé et al., 2003]. Ces modèles existent également en version adaptative, permettant de reproduire une plus grande variété de comportements en termes de séquences de spikes grâce au couplage d'une seconde équation différentielle [Izhikevich, 2003] [Izhikevich, 2004] [Brette and Gerstner, 2005] [Gerstner et al., 2014]. On peut finalement citer le modèle *Resonate-and-fire* qui est similaire à l'Integrate-and-Fire, mais fait intervenir des nombres complexes [Izhikevich, 2001]. Il intègre ainsi des oscillations sous-linaires du potentiel de membrane, ce qui le rend plus sensible aux temps précis des spikes et donc au codage temporel.

Le modèle Integrate-and-Fire est généralisé dans le modèle *SRM* (pour *Spike Response Model*) [Gerstner, 1995], pour lequel les paramètres dépendent non pas du potentiel, mais du temps écoulé depuis l'émission du dernier potentiel d'action. Ce modèle permet de modéliser plus précisément la période réfractaire. La variable d'intérêt est le potentiel de membrane, qui varie en fonction des signaux reçus dont l'effet dépend du temps écoulé depuis leur réception [Gerstner, 1995] [Maass and Bishop, 2001] [Gerstner and Kistler, 2002].

Revenons sur le modèle Hodgkin-Huxley. Il s'agit d'un modèle qui est plutôt impulsionnel que fréquentiel, car il s'intéresse explicitement aux potentiels d'action.

3.1.3 Modèles ponctuels versus modèles morphologiques

La plupart des modèles présentés jusqu'ici sont des modèles *ponctuels*, c'est-à-dire qu'il représentent le neurone comme un point, ignorant complètement sa morphologie. En particulier, les dendrites ne sont pas représentées bien qu'il soit largement admis qu'elles sont déterminantes dans le comportement du neurone (Chapitre 1, Section 1.1). C'est généralement la complexité des dendrites qui pousse les modélisateurs à les négliger.

Néanmoins, il existe des modèles qui se focalisent justement sur les dendrites. Le plus connu de tous est la *théorie des câbles appliquée aux dendrites* présentée dans les années 60 par *Wilfrid Rall*, pionnier de la modélisation des dendrites [Rall, 1959]. Elle se fonde sur une équation aux dérivées partielles de second ordre qui fut initialement proposée par *Lord (William Thomson) Kelvin* en 1855 [Thomson, 1856] [Holmes, 2013] pour décrire la propagation du signal électrique le long des câbles télégraphiques sous-marins. Cette théorie a été ensuite appliquée à la modélisation neuronale pour décrire la propagation du signal électrique dans les dendrites, assimilées à des câbles cylindriques.

En 1957, Rall a pointé du doigt une erreur dans l'estimation de paramètres électriques neuronaux qui était principalement due à une hypothèse impliquant l'absence de dendrites [Rall, 1957]. Ce travail préliminaire a souligné l'importance des propriétés dendritiques et c'est ce qui a motivé Rall à appliquer la théorie des câbles aux dendrites.

Dans sa version initiale, les dendrites sont décomposées en un ensemble de cylindres interconnectés, de longueur finie ayant des propriétés membranaires *uniformes* et exclusivement *passives* [Hodgkin and Rushton, 1946] [Davis and de No Lorente, 1947] [Rall, 1962]. L'hypothèse passive implique que les paramètres électriques soient constants. La distribution spatio-temporelle du potentiel le long d'un cylindre suit alors l'équation suivante, connue sous le nom de *cable equation* :

$$\frac{\partial U}{\partial T} = -U + \frac{\partial^2 U}{\partial^2 X} \quad (3.8)$$

où U est le potentiel membranaire, $T = t/\tau$ avec $\tau = RC$ une constante de temps et $X = x/\lambda$ est la distance parcourue avec une constante d'espace $\lambda = \sqrt{R/R_L}$, R étant la résistance membranaire, C la capacité membranaire et R_L la résistance longitudinale.

Il existe une solution générale à cette équation qui correspond à une somme infinie d'exponentielles décroissantes [Rall, 1969] [Stuart et al., 2016]. La recherche des solutions particulières qui dépendent notamment du point d'observation et du type de courant injecté a fait l'objet de nombreux travaux. La solution la plus simple est celle d'un câble infini en condition d'état stable (avec $\frac{\partial U}{\partial T} = 0$) :

$$U(x) = U_0 e^{-x/\lambda}$$

D'autres solutions sont décrites et notamment pour différentes conditions aux limites, dans les cas de câbles finis ou infinis [Rall and Rinzel, 1973] [Rinzel and Rall, 1974]

CHAPITRE 3. ÉTAT DE L'ART ET MOTIVATIONS

[Holmes et al., 1992] [Major et al., 1993] [Koch, 2004] [Rall, 2011]. Il existe de nombreux cas et la recherche de solutions est particulièrement fastidieuse.

La théorie des câbles propose également une approche *compartimentale*. Elle consiste à considérer les dendrites d'un neurone comme un arbre (au sens de la théorie des graphes) constitué de compartiments liés les uns aux autres : chaque compartiment est un câble ayant des propriétés électriques qui lui sont propres. Le potentiel est ainsi décrit par un large système d'équations différentielles dépendant du temps. Ce système peut être traité par les méthodes numériques standard et il a été montré que si les compartiments sont des « portions » de dendrites suffisamment petites, la solution du modèle compartimental converge vers celle du câble continu correspondant [Stuart et al., 2016]. Pour s'affranchir de l'hypothèse de propagation passive du signal, les compartiments peuvent être équipés de canaux ioniques non linéaires, permettant d'étudier numériquement l'effet de l'intégration active qui est bien plus complexe [Gerstner et al., 2014].

Sous certaines conditions, il n'est pas nécessaire de résoudre toutes ces équations. Rall a montré que l'intégralité de l'arbre dendritique peut être réduite à un unique *cylindre équivalent* dans le cas de certains types de courants injectés et si la structure obéit à la loi $d^{3/2}$, c'est-à-dire si le diamètre des branches à la puissance $3/2$ est égal à la somme des diamètres de leurs branches filles à la puissance $3/2$. Bien que les conditions soient pour le moins très restrictives, il s'agit d'une simplification largement exploitée [Rall, 1962] [Mel, 1994].

Ces travaux ont servi de cadre théorique pour comprendre comment les courants électriques se déplacent dans les dendrites [Koch et al., 1990] [Segev et al., 1994] [Koch and Segev, 1998]. Actuellement, ces théories sont exploitées notamment grâce à leur implémentation dans les environnements de simulation *Neuron* et *Genesis* [Hines and Carnevale, 1997] [Bower, 1998] [Stuart et al., 2016] [neuron, 2018] [genesis, 2018]. La simulation est un moyen d'étudier comment les différents signaux distribués sur l'arborescence dendritique interagissent dans le temps et l'espace pour déterminer les propriétés d'entrée/sortie des neurones. Malheureusement, ces systèmes restent très complexes et rendent difficile le passage à l'échelle multi neurones. C'est pourquoi des modèles plus simples incorporant des dendrites ont vu le jour.

Plusieurs travaux suggèrent que l'arborescence dendritique implémente des fonctions logiques booléennes. Un neurone unique peut alors agir comme un réseau multicouche de neurones formels : l'idée est que chaque neurone peut implémenter toute une hiérarchie d'opérations logiques dans ses dendrites [Koch et al., 1982] [Shepherd and Brayton, 1987]. L'idée sous-jacente est que les entrées sont groupées en sous-unités sur la base de leur proximité dans l'arbre dendritique et qu'elles interagissent de manière non linéaire. Cette vision est en faveur de la sensibilité au « cluster », c'est-à-dire que des synapses spatialement proches agissent en synergie lorsqu'elles sont co-activées (dans une fenêtre temporelle de quelques millisecondes). Ce phénomène, qui est soutenu par des travaux expérimentaux [Gasparini et al., 2004] [Katona et al., 2011] [Stuart et al., 2016], n'était pas mis en

exergue par la vision passive des dendrites. Pour aller plus loin dans l'exploration de ce type de comportements, des études furent menées sur le *clusteron* [Mel, 1992a]. Il s'agit d'un modèle de neurone abstrait constitué d'un soma sous la forme d'un neurone formel (au sens de McCulloch et Pitts) et d'un arbre dendritique qui reçoit des entrées pondérées (uniquement excitatrices). Le clusteron est muni d'une fenêtre glissante qui permet une interaction non linéaire entre les entrées proches : si deux entrées sont situées à une distance inférieure à celle de la fenêtre, elles se voient multipliées entre elles. Le soma effectue alors la somme des produits obtenus le long des dendrites et applique sa fonction d'activation. Ce modèle a notamment facilité l'exploration de règles de plasticité permettant de stocker de l'information dans la distribution des synapses [Mel, 1992a] [Mel, 1992b]. Le clusteron présente de nombreuses limitations parmi lesquelles : la prise en compte des synapses excitatrices uniquement, le traitement binaire du critère d'appartenance au cluster ainsi que l'hypothèse d'uniformité dendritique. Par exemple, le clusteron ne permet pas de prendre en compte la variation du diamètre des dendrites qui est un paramètre essentiel de l'intégration dendritique d'après la théorie des câbles [Stuart et al., 2016].

Pour poursuivre dans l'idée d'une hiérarchie d'opérations apportée par l'arborescence dendritique, le modèle *2LM* (pour *Two-layer Model*) [Poirazi et al., 2003] représente de manière explicite la structure multicouche. Le 2LM consiste en un soma autour duquel sont disposées, en étoile, des dendrites sans embranchements. Chaque dendrite est représentée par une sous-unité de type neurone formel généralement munie d'une fonction d'activation sigmoïde et le soma fournit une sortie à partir des sorties de chaque sous-unité dendritique indépendante. Bien qu'il repose sur de nombreuses hypothèses simplificatrices et qu'il impose d'importantes restrictions sur la structure des dendrites, ce modèle offre un cadre théorique pour l'étude du lien entre les opérations faites par les dendrites et celles effectuées à l'échelle des réseaux ainsi que pour l'analyse des capacités d'apprentissage [Poirazi and Mel, 2001] [Wu and Mel, 2009] [Stuart et al., 2016]. Il existe d'autres modèles qui représentent les dendrites actives à la manière de « petits neurones ». On peut notamment citer l'exemple de dendrites modélisées par des sous-unités de type Integrate-and-Fire [Gorski et al., 2017]. Ce travail souligne le rôle des propriétés dendritiques actives dans la fonction neuronale, mais ne prend pas en compte la structure précise des dendrites.

3.1.4 Modèles déterministes versus modèles probabilistes

Tous les modèles que nous avons présentés sont principalement fondés sur des équations différentielles et sont donc par essence *déterministes*. Il existe également des modèles probabilistes qui s'appuient sur des observations expérimentales. En effet, pour un même stimulus, la réponse d'un neurone est variable d'un essai à l'autre bien que la réponse moyenne converge généralement vers un motif spécifique. On ne détaillera pas ces modèles probabilistes ici, car la prise en compte des temps précis des spikes nécessaires à notre contexte, et le type d'approche que nous adoptons nous en éloignent d'emblée. On peut cependant citer : le modèle de *Siebert* (1965) qui représente les signaux par un processus de Poisson non homo-

gène et définit la probabilité d'émettre un potentiel en fonction du stimulus [Siebert, 1965], le modèle de *Nossenson* et *Messer* qui se présente sous forme d'une chaîne de Markov à deux états et permet de représenter la période réfractaire [Nossenson and Messer, 2010] et le modèle *Galves-Löcherbach* qui est un modèle probabiliste inspiré du type Leaky Integrate-and-Fire [Galves and Löcherbach, 2013].

Il existe également des modèles de neurones qui se fondent sur une équation différentielle stochastique. Par exemple, l'*équation de Langevin* intègre un terme de « bruit » au classique modèle Integrate-and-Fire [Gerstner and Kistler, 2002] pour tenir compte de l'arrivée stochastique des spikes.

3.2 Méthodes formelles pour les systèmes biologiques

Les *méthodes formelles* sont des techniques qui permettent d'étudier des systèmes de manière assistée par ordinateur, généralement grâce à une logique mathématique. Les méthodes formelles sont principalement utilisées pour la *spécification* et la *vérification* de systèmes électroniques ou informatiques critiques. La spécification formelle a l'avantage d'être précise et non ambiguë. Quant à la vérification, elle permet, par exemple, d'assurer l'absence de bug. Il existe différentes approches qui s'appuient notamment sur le *typage* [Pierce and Benjamin, 2002] (qui permet de vérifier partiellement que les programmes sont correctement écrits, mais qui éventuellement, par la correspondance de Curry-Howard, s'étend jusqu'à la preuve de théorèmes), l'*interprétation abstraite* [Cousot and Cousot, 1977] [Cousot, 2000] (qui permet d'établir des propriétés par sur-approximation des valeurs possibles des variables d'un système au cours du temps), la *vérification déductive* [Dijkstra, 1976] [Bobot, 2011] (qui permet de vérifier des propriétés en s'appuyant sur des inférences logiques) et le *model checking* (Chapitre 2, Section 2.4).

Les méthodes formelles peuvent également être utilisées pour d'autres applications. Le model checking présente un intérêt fort pour les systèmes biologiques dynamiques, car il permet de vérifier automatiquement si des modèles satisfont des *propriétés temporelles* particulières.

Dans cette section, nous donnons des exemples de coopération réussie entre les méthodes formelles et la biologie des systèmes, car notre objectif tend à étendre ce type d'approches aux neurosciences.

3.2.1 Le formalisme discret de René Thomas pour les réseaux génétiques

Contexte biologique

Un *réseau de régulation génétique* permet de modéliser l'effet des protéines régulatrices sur l'expression des gènes. De manière très simplifiée, un gène est une « portion » d'ADN qui peut être transcrite en ARNm, lui-même traduit en protéine. La protéine, qui est donc le

3.2. MÉTHODES FORMELLES POUR LES SYSTÈMES BIOLOGIQUES

produit du gène, peut alors réguler l'expression d'autres gènes en favorisant ou en inhibant leur transcription [Alberts et al., 2002]. Un gène qui favorise la transcription d'un autre gène est appelé *activateur* de ce gène et il est appelé *inhibiteur* lorsqu'il a l'effet inverse.

La dynamique des réseaux de régulation est complexe et peut être étudiée par différentes approches (équations différentielles, réseaux booléens, formalismes à base de règles, etc.) [De Jong, 2002].

Le formalisme de René Thomas

L'approche de *René Thomas*, formalisée par *Houssine Snoussi* [Snoussi, 1989], est une modélisation discrète des réseaux de régulations génétiques : le temps est discret et les états sont discrets également. Cette approche était à l'origine booléenne puis fut généralisée à une discrétisation multivaluée [Thomas, 1978] [Thomas and d'Ari, 1990] [Thomas, 1991] [Thomas et al., 1995]. L'espace des états est alors divisé en plusieurs « niveaux de concentration ». De plus, il existe une notion de *seuil* qui représente le niveau minimal à partir duquel la protéine peut agir sur ses gènes cibles. Le seuil, comme les niveaux de concentration possibles, est propre à l'entité considérée (on assimile le gène et sa protéine). Les interactions entre les différentes entités peuvent être représentées par un *graphe d'interaction* (Figure 3.6). Dans ce graphe, les sommets sont les entités d'intérêt et les arcs sont les influences entre ces entités. Les arcs sont étiquetés par le type de l'interaction (par un signe $+$ pour une activation et par un signe $-$ pour une inhibition) et par le seuil de l'entité représentée par le sommet source de cet arc.

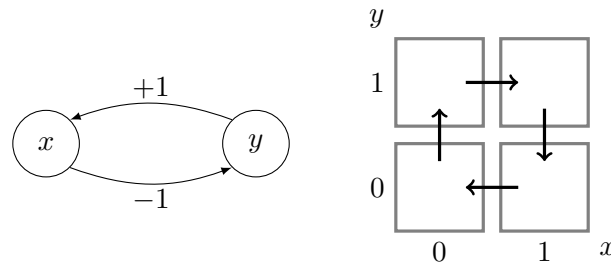


FIGURE 3.6 – Exemple simple d'un graphe d'interaction et de son graphe d'états (booléen) associés. L'entité y active l'entité x (signe $+$) lorsqu'elle est au niveau 1 et en retour l'entité x inhibe l'entité y (signe $-$) lorsqu'elle est au niveau 1. On peut voir sur le graphe d'état que les entités x et y peuvent être soit au niveau 0, soit au niveau 1. Prenons par exemple l'état initial $(0,1)$ soit x au niveau 0 et y au niveau 1, ce qui correspond à la case en haut à gauche du graphe d'états. Puisque y atteint son seuil (avec des paramètres « raisonnables » que nous passons sous silence ici), il active x qui tend alors vers 1 soit vers l'état $(1,1)$. Puis, x atteignant son seuil, il peut alors inhiber y en retour qui repasse à 0, menant à l'état $(1,0)$. Enfin, x étant à un niveau trop faible pour agir sur y , le système retourne vers l'état $(0,1)$ et ainsi de suite. Le graphe d'états représente ainsi une dynamique oscillatoire.

À partir du graphe d'interaction, le formalisme de René Thomas permet d'étudier la *dynamique* du réseau, c'est-à-dire l'évolution temporelle du niveau de chaque entité. Cette

dynamique est caractérisée par des *paramètres*. Chaque paramètre représente le point focal, c'est-à-dire le niveau vers lequel tend un gène en fonction de ses régulateurs effectifs (dont le niveau est au-dessus du seuil). Naturellement, en présence de son activateur, un gène cible tendra vers un niveau plus élevé que lorsque l'activateur est absent ou lorsqu'un inhibiteur est présent. La dynamique est représentée dans le *graphe d'états* qui est simplement un automate (Figure 3.6).

Il est important de préciser que l'approche discrète de René Thomas est capable de capturer les principales caractéristiques de la dynamique continue [Snoussi, 1989] [Thomas and Kaufman, 2001]. De par son haut niveau d'abstraction, l'approche discrète a été instrumentée par des méthodes formelles provenant de l'informatique et notamment par la puissance du model checking [Bernot et al., 2004].

Apport des méthodes formelles

Les paramètres dynamiques d'un réseau de régulation génétique sont généralement complexes à identifier. Néanmoins, ils peuvent être déterminés grâce aux observations partielles et aux hypothèses des biologistes. Par exemple, des expériences peuvent faire apparaître des états stables, des oscillations ou des successions d'états. Ces observations peuvent être exprimées en formules logiques temporelles, permettant alors l'utilisation du model checking. La logique *CTL* (pour *Computation Tree Logic*) est particulièrement adaptée, car le graphe d'état (asynchrone) adopte une structure arborescente par rapport au temps [Emerson, 1990] [Huth and Ryan, 2004].

Exemple: En *CTL*, on peut exprimer des propriétés temporelles telles que :

- $EF\phi$ qui signifie « il existe un état atteignable dans lequel la propriété ϕ est satisfaite » ;
- $AG(\phi \rightarrow E[\phi U \psi])$ qui signifie « à partir de tous les états atteignables satisfaisant la propriété ϕ , il existe un chemin qui maintient ϕ jusqu'à atteindre un état où la propriété ψ est satisfaite ».

Concernant la syntaxe, E vient de *Exists* et signifie « il existe un chemin », F vient de *Future* et signifie « dans des états futurs », A vient de *All* et signifie « pour tous les chemins », G vient de *Globally* et signifie « dans tous les états futurs », U vient de *Until* et signifie « jusqu'à un état ».

Le model checking permet alors de vérifier qu'un modèle satisfait une propriété donnée, permettant ainsi de sélectionner les modèles compatibles avec la propriété. En procédant ainsi, la formalisation permet d'orienter les plans d'expérience des biologistes, qui peuvent permettre d'améliorer le modèle en retour [Bernot et al., 2004].

Cette méthode est implémentée dans l'outil *SMBioNet* [Richard, 2010] qui permet d'inférer les paramètres dynamiques en fonction de connaissances biologiques exprimées en *CTL*. Cet outil a notamment permis d'étudier le *switch génétique* de la bactérie pathogène *Pseudomonas aeruginosa*, ouvrant des perspectives thérapeutiques [Filopon et al., 2006].

De nombreuses améliorations et approches complémentaires ont été développées. Elles permettent notamment une meilleure efficacité du model checking [Bernot et al., 2008] [Fromentin et al., 2007] [Gallet, 2016]. De plus, le cadre de modélisation de René Thomas a été étendu pour permettre l'intégration d'un temps continu au sein des états discrets [Behaegel et al., 2016] [Cornillon, 2017]. La représentation sous forme d'automate hybride conserve un niveau d'abstraction suffisant pour bénéficier des méthodes formelles dans l'identification des paramètres dynamiques [Behaegel et al., 2016] [Behaegel et al., 2018].

3.2.2 Les réseaux de Petri pour les réseaux métaboliques

Les réseaux de Petri

Les *réseaux de Petri*, inventés par *Carl Adam Petri* en 1962 sont un outil graphique et mathématique permettant de modéliser, de simuler et de vérifier des systèmes à variables discrètes. Un réseau de Petri est défini par un ensemble de *places*, un ensemble de *transitions* et un ensemble d'*arcs* liant les places et les transitions (un arc ne doit jamais lier deux places ou deux transitions). Il existe un *marquage* initial correspondant à un nombre entier de *jetons* dans chaque place. Enfin, chaque arc est étiqueté par le nombre de jetons consommés depuis une place vers une transition ou arrivant sur une place par une transition (Figure 3.7).

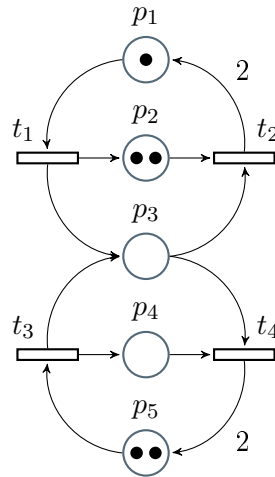


FIGURE 3.7 – Exemple de réseau de Petri. Les rectangles sont les transitions, les cercles sont les places dans lesquelles peuvent se trouver des jetons (points noirs). Un arc sortant d'une transition étiquetée par le chiffre 2 signifie que le passage de cette transition consomme 2 jetons. Lorsque l'arc n'est pas étiqueté, le nombre de jetons consommés vaut 1.

Les transitions représentent des événements qui ne peuvent avoir lieu que lorsque les places prédécesseuses contiennent assez de jetons (*ressources*). Par exemple, dans la Figure 3.7, la transition t_4 ne peut être *franchie* car elle nécessite deux ressources qui ne sont pas présentes dans les places prédécesseuses (p_3, p_4) de cette transition. Parmi les transitions *franchissables*, une transition est *tirée* ce qui modifie le marquage c'est-à-dire l'état du réseau. Les jetons (dont le nombre dépend de l'étiquetage des arcs) se déplacent alors des

places prédécesseurs vers les places successeurs de la transition tirée. Par exemple, dans la Figure 3.7, si la transition t_3 est tirée, la place p_5 n'aura plus qu'un jeton et les places p_3 et p_4 en contiendront un également.

Les réseaux de Petri sont très utilisés dans la conception et la vérification de logiciels, mais ils peuvent être employés pour bien d'autres applications et notamment pour la modélisation de systèmes biologiques [Heiner and Koch, 2004].

Modélisation de réseaux métaboliques

Les réseaux de Petri sont particulièrement adaptés pour modéliser des *réseaux métaboliques*, c'est à dire des ensembles de réactions biochimiques en interaction (Figure 3.8). Les transitions représentent les réactions et les places représentent alors les composés qui participent aux réactions. Les jetons indiquent la présence des composés et l'étiquetage des arcs représente les coefficients stœchiométriques des réactions. L'orientation des arcs dépend donc de la faisabilité thermodynamique des réactions [Reddy et al., 1993].

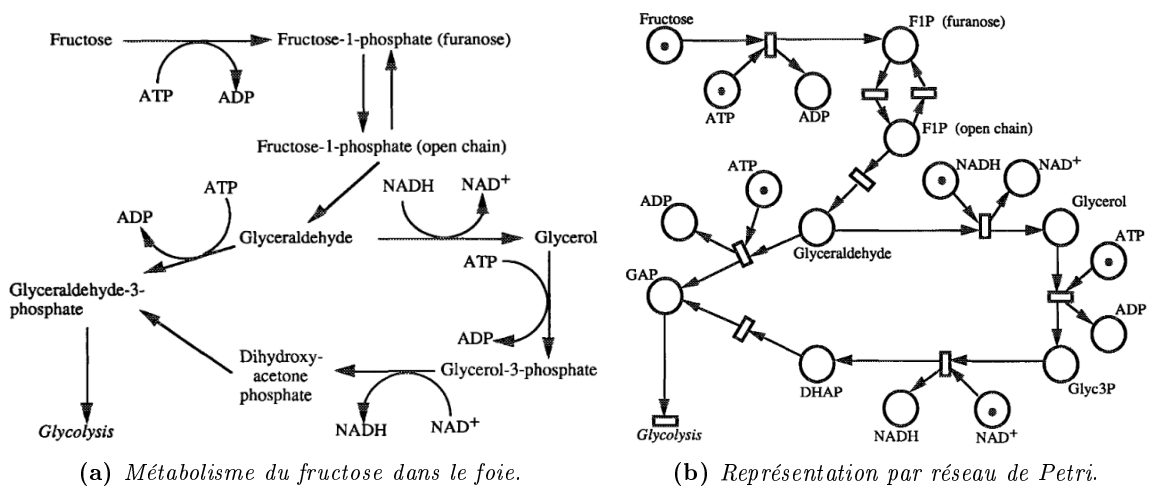


FIGURE 3.8 – Métabolisme du fructose dans le foie et sa représentation sous forme de réseau de Petri. Figure extraite de [Reddy et al., 1993].

Les réseaux de Petri bénéficient d'un grand nombre de techniques d'analyse et de vérification. La vérification permet notamment de valider que le modèle a bien le comportement attendu, avant de l'exploiter pour répondre à des questions plus sophistiquées. Dans ce but, on peut s'intéresser aux propriétés fondamentales du réseau et en particulier aux invariants de transitions (*T-invariants*) et de places (*P-invariants*) [Reddy et al., 1993] [Heiner and Koch, 2004] [Bourcerie, 2010]. Les T-invariants sont des ensembles de transitions qui reproduisent un marquage et donc un état donné. Dans le contexte des réseaux métaboliques, les T-invariants représentent un ensemble de réactions qui maintiennent le système dans un état stationnaire. Ainsi, si toutes ces réactions se produisent, l'état du système est inchangé. Quant aux P-invariants, ils correspondent à des règles de conservation

3.2. MÉTHODES FORMELLES POUR LES SYSTÈMES BIOLOGIQUES

de la matière [Comet, 2014]. Le calcul des invariants, qui peut être fait automatiquement par des manipulations formelles, requiert uniquement un raisonnement structural qui ne nécessite donc pas de générer l'espace des états [Heiner and Koch, 2004]. Il existe différents outils pour analyser des systèmes formalisés en réseaux de Petri. En particulier, *INA* (*Integrated Net Analyzer*) permet non seulement de calculer les invariants, mais aussi de répondre à des questions relatives à l'atteignabilité, la nature bornée, la vivacité ou la réversibilité [Starke, 2000]. Cet outil permet également de vérifier des propriétés en CTL.

L'approche par réseaux de Petri a permis d'étudier différents systèmes biologiques [Heiner and Koch, 2004]. C'est une approche qui a par exemple été utilisée pour modéliser et analyser le métabolisme de *Solanum tuberosum* (pomme de terre) qui représente un important enjeu agronomique [Koch et al., 2004].

Des extensions temporelles des réseaux de Petri ont également été développées, s'adaptant particulièrement à la modélisation des systèmes biologiques. Des outils tels que *TINA* [LAAS, 2018], *WoPeD* [WoPeD, 2018] et *Romeo* [IRCCyN, 2018] permettent de les étudier. On peut citer le cas particulier des *réseaux de Petri hybrides fonctionnels* qui possèdent une très grande expressivité [Ahmed et al., 2017], mais dont le pouvoir d'expressivité compromet la vérification automatique [Troncale et al., 2006] [Comet, 2014].

3.2.3 Les formalismes à base de règles

Les systèmes biologiques peuvent être modélisés par des formalismes à base de *règles*. Le plus représentatif, *BIOCHAM* [Fages et al., 2004] [Calzone et al., 2006a] [inria saclay-ile de france, 2018] (pour *BIOCHemical Abstract Machine*), est un environnement permettant la formalisation, la simulation et l'analyse de systèmes biologiques. *BIOCHAM* permet également de vérifier automatiquement qu'un modèle satisfait des propriétés biologiques ou d'inférer des valeurs de paramètres. Ce formalisme se fonde ainsi sur deux langages :

- un langage simple à base de règles pour décrire le système qui consiste généralement en un réseau de réactions biochimiques ;
- un langage basé sur la logique temporelle pour exprimer des propriétés à partir d'expériences biologiques.

Les règles *BIOCHAM* représentent des réactions entre des entités qui peuvent être des petites molécules chimiques ou biochimiques, mais également des protéines ou des gènes. De manière simplifiée, la syntaxe est la suivante :

```
objet = molécule | molécule :: localisation
molécule = nom | molécule-molécule | molécule~{nom, . . . , nom}
réaction = composés => composés
composés = _| objet | k*objet | composés + composés
```

L'objet de base est la molécule et l'opérateur $::$ permet de préciser sa localisation (par exemple le noyau ou le cytoplasme). L'opérateur $-$ permet de représenter des complexes de molécules et l'opérateur \sim permet de représenter des modifications de la molécule (par exemple une phosphorylation). Par exemple, l'objet $X\{p\}-Y::\text{nucleus}$ représente un complexe entre la molécule X phosphorylée et la molécule Y , dans le noyau de la cellule. Une règle de réaction décrit la transformation d'un ensemble de composés en un autre. Ainsi, la règle $X = [E] \Rightarrow Y$ correspond à la réaction catalysée par E qui transforme X en Y soit la réaction $X + E \Rightarrow Y + E$. Il est également possible d'ajouter aux réactions des paramètres cinétiques k [Calzone et al., 2006a].

À l'origine, BIOCHAM était restreint à des modèles booléens. Actuellement, il permet d'effectuer des simulations continues et stochastiques [Calzone et al., 2006b]. Le langage offert par BIOCHAM est un langage riche, doté d'un fort pouvoir d'expression, qui permet d'étudier des modèles relativement complexes constitués de centaines de règles et de variables. Par exemple, des propriétés CTL ont été testées par model checking sur le modèle du cycle cellulaire présenté dans [Kohn, 1999] [Calzone et al., 2006a] qui implique 800 règles et 500 variables.

Il existe également d'autres formalismes à base de règles similaires à BIOCHAM tels que *Pathway Logic* [Eker et al., 2004] qui est particulièrement adapté à la modélisation des voies de signalisation biologiques, *Bioambients* [Regev et al., 2004] qui accorde une importance particulière à la compartimentation des espèces ou encore le cadre de modélisation dédié au domaine de la toxicologie, introduit dans [Miraglio et al., 2016], qui se focalise sur les équilibres.

3.2.4 Méthodes formelles pour les neurosciences

Il existe très peu d'applications des méthodes formelles aux neurosciences. On peut tout de même citer les travaux récents de *Franck Grammont, et. al* qui sont pertinents pour ce domaine d'application.

Vérification de propriétés d'archétypes

Dans l'article [De Maria et al., 2016], les auteurs présentent une version discrète du modèle Leaky Integrate-and-Fire (Section 3.1). Nous adopterons une approche similaire dans le chapitre 7.

Dans la version classique du Leaky Integrate-and-Fire, les états et le temps sont potentiellement infinis. Le potentiel de membrane p , à un temps $t \in [0, +\infty]$, intègre les entrées reçues à cet instant, ainsi que ce qui reste des entrées reçues dans le passé (atténuées par la fuite). Le calcul du potentiel de membrane peut donc être approximé par l'expression suivante :

$$p(t) = \sum_{e=0}^{+\infty} \sum_{j=1}^m r^e x_j(t - e)$$

3.2. MÉTHODES FORMELLES POUR LES SYSTÈMES BIOLOGIQUES

où $e \in \mathbb{R}^+ \cup +\infty$ représente le temps écoulé depuis l'émission du dernier spike par le neurone d'intérêt, r est le « coefficient de reste » et les x_j sont les entrées. Pour des raisons de simplicité, tous les poids synaptiques sont supposés égaux à 1.

Puisque l'effet des entrées décroît selon une loi de puissance avec le temps, ils admettent que les entrées reçues depuis un temps suffisamment long peuvent être négligées. Ils définissent donc une fenêtre temporelle d'intégration σ qui dépend d'un seuil d'erreur arbitraire ϵ . Ainsi, plus ϵ est grand et plus σ est petit. Le temps duquel dépend le potentiel n'est alors plus infini, mais borné à $[t - \sigma, t]$. En discrétisant le temps, le potentiel peut être défini par une simple somme :

$$p(t) = \sum_{e=0}^{\sigma} r^e \sum_{j=1}^m x_j(t - e)$$

Chaque temps t peut alors être considéré comme une transition et le calcul du potentiel dépend dorénavant d'une mémoire finie.

La version discrète du Leaky Integrate-and-Fire, caractérisée par uniquement 3 paramètres (le coefficient de reste, la fenêtre d'intégration et le seuil d'activation), peut alors être facilement implémentée en Lustre.

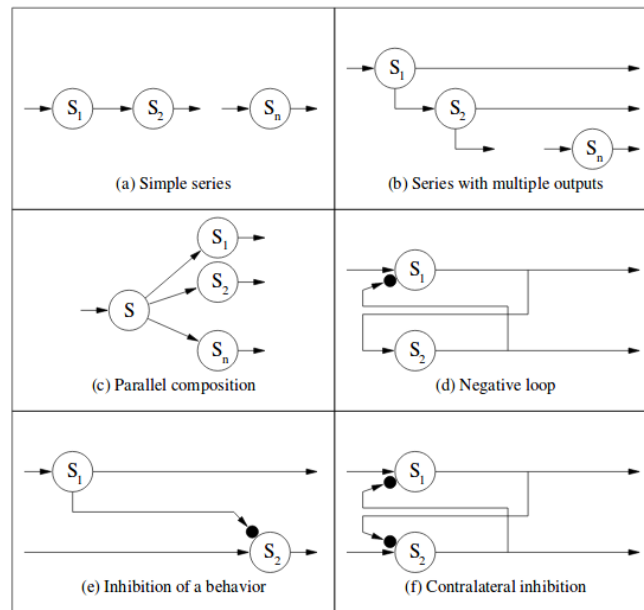


FIGURE 3.9 – Les archétypes sont des graphes constitués de quelques neurones, ayant un comportement d'intérêt. Ces petits graphes peuvent être concaténés de différentes façons pour former de plus gros circuits. La série simple (a) est une séquence de neurones en chaîne, la série à sorties multiples (b) est une série dans laquelle la sortie de chaque neurone est considérée, la composition parallèle (c) est un ensemble de neurones recevant comme entrées la sortie d'un neurone commun, la boucle négative (d) contient deux neurones dont l'un qui active l'autre qui lui-même inhibe le premier neurone en retour, l'inhibition de comportement (e) consiste en un neurone inhibé par un autre et l'inhibition controlatérale (f) est constituée de neurones qui s'inhibent mutuellement. **Figure extraite de [De Maria et al., 2016].**

Les auteurs ont, à partir de cet élément de base, encodé de petits circuits neuronaux sur

lesquels ils ont vérifié des propriétés par model checking (Figure 3.9). Ces petits circuits constituent l'un des éléments de base d'une théorie de l'organisation et du développement du système nerveux, développée par Franck Grammont. Ils représentent des archétypes c'est-à-dire des graphes constitués de quelques neurones, ayant une structure et un comportement d'intérêt sur le plan neurophysiologique. Il s'agit de comprendre quels sont les comportements élémentaires permis par de petits circuits de neurones (archétypes) que l'on peut retrouver au niveau des structures nerveuses les plus primitives sur le plan phylogénétique (moelle épinière, tronc cérébral...). Ces archétypes peuvent également être combinés pour dupliquer les fonctions ou, au contraire, pour former des circuits et des fonctions plus complexes. Les propriétés de ces archétypes ont également un intérêt sur le plan computationnel, informatique ou électronique. Ces propriétés peuvent par exemple être exploitées dans une perspective de réduction de réseaux : sur la série simple (Figure 3.9), ils ont montré de manière automatique que deux séries de longueurs différentes et ayant des paramètres différents peuvent avoir le même comportement, quelle que soit l'entrée. Avec des paramètres adaptés (seuil d'activation et poids synaptiques), il est notamment possible de réduire une série constituée de 4 neurones à une série de 3, en conservant le même comportement.

Modélisation de neurones par automates temporisés

Dans l'article [De Maria et al., 2017], les auteurs proposent une formalisation du modèle Leaky Integrate-and-Fire (Section 3.1) par *automate temporisé* [Alur and Dill, 1994].

Les automates temporisés constituent un puissant outil théorique particulièrement adapté à la modélisation et à la vérification de systèmes temps réels. De manière simplifiée, un automate temporisé est un graphe orienté muni d'un ensemble d'horloges qui avancent toutes au même rythme. Les nœuds (ou états) du graphe sont associés à des *invariants* qui constituent les conditions pour rester ou entrer dans l'état. Les arcs sont équipés de *gardes* qui imposent principalement des contraintes sur les horloges pour passer d'un état à un autre. Il est possible de synchroniser plusieurs automates temporisés pour qu'ils fonctionnent en parallèle.

Les auteurs définissent un neurone comme un automate temporisé à 3 états (Figure 3.10).

- *A* (pour *Accumulation*) est un état dans lequel le neurone intègre les entrées qu'il reçoit. Le neurone reste dans cet état tant que le temps t ne dépasse pas une période d'accumulation donnée T . Pendant ce temps, lorsque le neurone reçoit des spikes, il stocke dans une variable a la valeur de la somme pondérée de ces entrées à ajouter à son potentiel p .
- *D* (pour *Decision*) est l'état dans lequel passe le neurone lorsque t est supérieur à T . Le coefficient de fuite est appliqué à la valeur du potentiel et la valeur de a est ajoutée pour prendre en compte les entrées reçues lors de la précédente période d'accumulation. Dans cet état, le temps ne s'écoule pas. Le neurone peut alors retourner dans l'état *A* ou passer dans l'état *W* si le potentiel atteint le seuil.

- W (pour *Wait*) est l'état dans lequel se trouve le neurone en période réfractaire (de durée τ). En entrant dans cet état, le neurone émet un spike et attend. Il peut ensuite retourner dans l'état A , réinitialisant p , a et t .

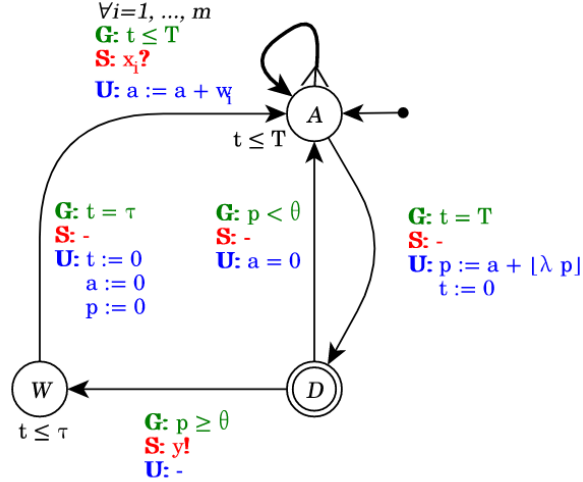


FIGURE 3.10 – Automate temporisé pour modéliser un neurone de type *Leaky Integrate and Fire*. La garde G est relative à l'horloge, la garde S est relative aux spikes et la garde U concerne le potentiel. La variable t est le temps qui s'écoule, T est la période d'accumulation, les i représentent les entrées du neurone, a est une variable de stockage, p est la valeur du potentiel, λ est le coefficient de fuite, θ est la valeur du seuil et τ est la durée de la période réfractaire. *Figure extraite de [De Maria et al., 2017].*

Les auteurs ont utilisé l'outil *Uppaal* [uppaal, 2015] pour analyser ce modèle. Uppaal est un environnement intégré qui permet notamment de vérifier des propriétés par model checking. Grâce à cet outil, ils ont montré de manière automatique que leur automate temporisé préserve les comportements typiques du modèle *Leaky Integrate-and-Fire* définis par Izhikevich [Izhikevich, 2004].

3.3 Nos motivations

Comme nous l'avons vu dans ce chapitre, les méthodes formelles ont eu un apport notable dans différentes branches de la biologie. Notre motivation première est d'investiguer l'apport des méthodes formelles aux neurosciences. Nous précisons, dans cette section, le contexte dans lequel nous avons commencé cette investigation.

Dans cette thèse, nous souhaitons modéliser un neurone unique adapté à l'étude les assemblées de neurones (Chapitre 1, Section 1.3).

Rappelons qu'une assemblée est généralement définie comme un groupe de neurones qui répondent ensemble à un stimulus particulier, permettant de le représenter. C'est la synchronisation des potentiels d'action émis par les neurones qui caractérisent leur appartenance à une assemblée. Depuis son postulat par Hebb, la théorie des assemblées a été étayée par de nom-

CHAPITRE 3. ÉTAT DE L'ART ET MOTIVATIONS

breuses observations [Gray, 1999]. Cependant, la nature exacte et la dynamique des assemblées ne sont actuellement pas complètement établies, principalement en raison des limites expérimentales qui persistent [Buzsáki, 2004] [Meunier et al., 2006] [Malvache et al., 2016] [Holtmaat and Caroni, 2016].

La principale difficulté technique de l'étude expérimentale des assemblées provient de la nécessité d'enregistrer plusieurs neurones simultanément. En effet, la coordination temporelle entre les potentiels d'action des différents neurones ne peut être observée qu'à l'échelle d'une population [Harris, 2005]. Un autre problème avec l'identification des assemblées est leur nature transitoire. La durée d'activation des assemblées serait de l'ordre de quelques dizaines à quelques centaines de millisecondes au maximum, ce qui augmente le risque de passer à côté des périodes de coordination [Singer et al., 1997]. Enfin et surtout, l'observation expérimentale des assemblées ne peut se faire qu'une fois établies, ce qui ne permet pas d'étudier leur mise en place.

Aux limites relatives à l'enregistrement de l'activité des neurones, s'ajoute l'interprétation délicate des données. Tout d'abord, les données pertinentes doivent être extraites d'un signal largement bruité [Lewicki, 1998] [Riehle et al., 2000]. En particulier, lorsque les expériences se font sur un organisme engagé dans une tâche comportementale, les mesures électrophysiologiques acquises peuvent être particulièrement bruitées par les mouvements et les activités involontaires. Puis, il est nécessaire de faire appel à des méthodes statistiques élaborées pour éventuellement établir des corrélations temporelles entre les différents signaux et ainsi révéler des assemblées [Grün et al., 1999] [Gerstein and Kirkland, 2001] [Reynaud-Bouret et al., 2014] [Tuleau-Malot et al., 2014]. Le principe général de ces méthodes est d'évaluer si les synchronisations observées se produisent significativement plus que celles attendues par chance (sur la base des fréquences d'émission des neurones) [Grammont and Riehle, 1999] [Riehle et al., 2000]. Par ailleurs, l'absence de données de connectivité anatomique complexifie d'autant plus l'analyse.

Au vu de la complexité de l'étude expérimentale, le recours à l'étude théorique apparaît alors complémentaire et nécessaire pour tester des hypothèses sur les assemblées, dans une perspective non plus seulement descriptive, mais également explicative. En particulier, l'approche *in silico* est le seul moyen d'étudier les mécanismes à l'origine de leur mise en place.

L'approche *in silico* classique repose sur de la simulation de modèles de neurones uniques mis en réseaux. Nous avons vu dans la section 3.1 de ce chapitre qu'il existe différents types de modèles. Les environnements de simulation les plus utilisés en neurosciences computationnelles, tels que *Neuron* ou *Genesis*, implémentent des modèles biologiquement plausibles, comme le modèle Hodgkin-Huxley (éventuellement combiné à la théorie des câbles). Ce type de modèles permet ainsi de simuler le comportement des neurones de manière très précise. En contrepartie, leur complexité intrinsèque implique des ressources et des temps de calcul particulièrement importants lors du passage à l'échelle des réseaux requise pour l'étude des assemblées. Ce type d'environnement implique également de traiter un grand nombre d'hypothèses sur le plan biologique afin d'ajuster les nombreux paramètres du modèle. De plus,

elle rend difficile leur analyse.

Il existe également des modèles, plus simples, basés sur l'approximation des fonctions neuronales par des équations à peu de paramètres. L'avantage de ce type de modèles est de faciliter la mise en réseaux, nécessaire à l'étude des assemblées. Ces modèles sont, cependant, bien trop simplistes. L'aspect temporel étant essentiel dans le fonctionnement des assemblées, un modèle de type McCulloch et Pitts, par exemple, est beaucoup trop abstrait pour être utilisé dans ce contexte.

Le modèle Leaky Integrate-and-Fire qui a déjà servi de base pour des études formelles (Section 3.2) nous semble intéressant de par l'aspect temporel qu'il intègre à un niveau d'abstraction intermédiaire. Un autre avantage de ce modèle est d'être de type impulsif : le codage temporel sur lequel il se fonde, qui est de plus en plus soutenu par les observations expérimentales, est particulièrement adapté à l'étude des phénomènes de synchronisation.

Cependant, comme la plupart des modèles abstraits, le Leaky Integrate-and-Fire est un modèle ponctuel qui ne tient, par définition, pas compte de la morphologie des neurones. Or, comme nous l'avons vu dans le Chapitre 1, le comportement des neurones biologiques dépend largement des propriétés biophysiques des dendrites et de leur morphologie. Plus précisément, les dendrites conditionnent l'intégration spatio-temporelle des potentiels post-synaptiques par les neurones [Williams and Stuart, 2002] [Stuart et al., 2016].

À l'échelle des réseaux, la dimension temporelle apportée par la morphologie des neurones serait également déterminante dans la coordination des activités, caractéristique des assemblées [Izhikevich, 2006]. Les modèles ponctuels, qui négligent ces phénomènes, sont par conséquent fondamentalement incompatibles avec l'étude des assemblées, même s'ils ont été malgré tout utilisés pour ce faire [Diesmann et al., 1999].

Parmi les modèles de neurones uniques existants, il n'en existe, à notre connaissance, pas qui soient satisfaisant dans le contexte biologique où nous nous plaçons. Notre objectif dans cette thèse est de combler ce manque en proposant un modèle de neurone qui puisse être exploité pour l'étude des assemblées de neurones. C'est un modèle qui doit donc être aussi plausible que possible du point de vue neurophysiologique et en termes de *traitement de l'information neuronale*, en incorporant notamment des dendrites. De plus, il doit être suffisamment abstrait, car on souhaite bénéficier de propriétés algébriques utiles pour tirer profit, à terme, des méthodes formelles de l'informatique.

Nous avons vu dans la section précédente que les méthodes formelles peuvent être précieuses pour l'étude des systèmes biologiques dynamiques. Au-delà des approches centrées sur la simulation, les méthodes formelles permettent de *prouver automatiquement* des propriétés du système considéré. Le succès des méthodes formelles, et en particulier du model checking, est favorisé par des abstractions judicieuses, en conjonction avec des propriétés mathématiques exploitables. Par exemple, la théorie de René Thomas a été formalisée, ce qui a permis d'exprimer la sémantique des réseaux biologiques sous forme d'automate et de faire du model checking en exprimant les propriétés biologiques en logique temporelle. La formalisation

s'est effectuée dans des conditions favorables, car René Thomas et Houssine Snoussi avaient montré qu'il était possible de découper l'espace des phases en un nombre fini de sous-espaces de comportement uniforme. De manière complémentaire, dans le cas des réseaux de Petri, l'identification des invariants peut être obtenue par des méthodes formelles fondées sur les propriétés d'algèbre matricielle du formalisme.

En collaboration avec *Olivier Roux* et *Morgan Magnin* du *Laboratoire des sciences du numérique de Nantes (LS2N)* [ls2n, 2018], nous avons investigué, sans succès, la possibilité de développer un modèle de neurone avec des formalismes existants. Les réseaux de Petri, comme les automates hybrides, ont déjà fait leurs preuves pour la modélisation de systèmes biologiques, et bénéficient aujourd'hui de nombreux outils, permettant notamment des preuves automatiques (Section 3.2). L'automate hybride est particulièrement adapté au soma du neurone à la manière d'un Leaky Integrate and Fire, puisqu'il permet de modéliser une variation continue du potentiel à l'intérieur d'un état discret, qui pourrait dépendre du statut du neurone (en période réfractaire absolue/relative ou non). Cependant, lorsqu'il s'agit d'ajouter des dendrites, l'automate hybride ne convient plus. En effet, il n'est pas possible de gérer en parallèle les états dans lesquels, d'une part, la synapse attend l'arrivée d'un spike, et d'autre part, l'état dans lequel la synapse transfère au soma la contribution d'un spike antérieur. Au contraire, les réseaux de Petri semblent adaptés lorsque l'on choisit de modéliser le signal par des jetons qui se déplacent dans le réseau, car ce formalisme gère parfaitement la concurrence dans les dendrites. Nous n'avons néanmoins pas choisi ce formalisme, car la modélisation du soma provoquerait l'explosion des états, surtout à cause de la fuite. Effectivement, la fuite du signal que l'on retrouve dans le modèle Leaky Integrate-and-Fire n'est pas une quantité fixe retirée au potentiel, mais une quantité qui dépend de la valeur courante du potentiel et ceci entraîne la démultiplication des places correspondant au soma. Finalement, ces échecs ne sont pas très surprenants, puisque l'on a besoin de gérer une quantité de charges en tout point de la structure du neurone, c'est-à-dire que l'on considère une infinité de nombres réels. Or, par définition, les automates hybrides (et *a fortiori* les réseaux de Petri) ont un nombre fini d'états, chaque état pouvant porter un nombre fini de réels (ou d'entier). Si l'on devait accepter un nombre infini de réels, nous perdriions tous les avantages, en termes d'outils, de ces formalismes.

Pour répondre à notre objectif, nous avons donc choisi de définir un modèle de neurone *ad hoc* compatible avec la biologie, à un niveau d'abstraction juste suffisant pour bénéficier de propriétés algébriques, ouvrant la porte au raisonnement mathématique assisté par ordinateur.

Deuxième partie

NEURONES ABSTRAITS AVEC DENDRITES : NOTRE CADRE DE MODÉLISATION

Cette partie a pour objectif d'introduire notre cadre de modélisation. Nous introduisons un modèle de neurone formel avec dendrites et présentons ses propriétés mathématiques fondamentales.

Le dernier chapitre de cette partie est dédié à une vision purement algébrique du formalisme.



CHAPITRE 4

DÉFINITION DU CADRE DE MODÉLISATION

Notre cadre de modélisation est *hybride* en ce sens qu'un neurone reçoit des signaux d'entrée discrets aux synapses puis les convertit en un signal continu qu'il rediscrétise en sortie du soma. Les signaux discrets représentent la suite des potentiels d'action, que l'on appelle généralement *spikes*, en modélisation. Par conséquent, il s'agit d'un hybride différent de la notion d'hybride classique des automates (Chapitre 3, Section 3.2). Ce qui nous intéresse dans ce travail est la fonction d'entrée/sortie, c'est-à-dire la relation entre les spikes d'entrée reçus par un neurone et les spikes émis en sortie par celui-ci (via ce passe intermédiaire continu). Nous adoptons de plus une vision très abstraite qui ne descendra pas jusqu'aux considérations moléculaires.

Pour rappel et de manière simplifiée, un neurone biologique reçoit des potentiels d'action aux synapses. Ceci génère des courants électriques, plus précisément des mouvements d'ions, faisant varier localement et transitoirement le potentiel de membrane. Cette variation est appelée potentiel post-synaptique. Les courants se déplacent ensuite dans les dendrites jusqu'à atteindre le soma, propageant ainsi la variation du potentiel de membrane. Le soma accumule toutes les charges et lorsque le potentiel de membrane résultant atteint un certain seuil, le neurone émet un potentiel d'action.

Habituellement, en biologie comme en modélisation, la grandeur d'intérêt est le potentiel de membrane, c'est-à-dire la différence de voltage (ou tension) entre l'intérieur et l'extérieur du neurone. Ceci est une conséquence directe des techniques de mesure expérimentales disponibles. Ici, nous faisons le choix original de considérer une notion abstraite de *charge électrique intracellulaire locale* (que l'on appellera simplement *charge*), plutôt que les valeurs habituelles de potentiel. Nous choisissons, pour des raisons de simplicité, de normaliser la charge à zéro quand le potentiel de membrane est à sa valeur de repos. Lorsque le potentiel de membrane est perturbé, la valeur de la charge représente le bilan local des charges (positives et négatives) portées par les ions intracellulaires à l'origine de la perturbation.

CHAPITRE 4. DÉFINITION DU CADRE DE MODÉLISATION

Comme nous allons le voir dans ce chapitre, l'avantage de s'intéresser à la charge est de pouvoir faire des *sommes* aux embranchements dendritiques, ce qui permet une simplification drastique des définitions. En effet, lorsque les charges se propagent dans les dendrites, elles se rencontrent aux embranchements de l'arborescence ; il apparaît alors naturel de sommer ces charges pour obtenir une valeur de charge électrique locale à l'embranchement. Cette opération ne serait en revanche pas crédible si l'on considérait des tensions.

Une alternative serait *a priori* de considérer des *intensités* de courant. Effectivement, les intensités pourraient aussi être sommées aux embranchements. Cependant, nous souhaitons pouvoir relier facilement notre grandeur d'intérêt à la tension qui est le standard des données biologiques et le choix des intensités serait alors inadéquat comme démontré ci-dessous.

Aux bornes d'une résistance (R), la tension (U) et l'intensité (I) sont proportionnelles, de par la fameuse *loi d'Ohm* : $U = R \times I$. Or, la membrane du neurone est généralement modélisée comme un circuit électrique qui comprend non seulement une résistance, qui représente les canaux ioniques, mais également un condensateur, qui représente les propriétés isolantes et polarisables de la membrane (Figure 4.1 et Chapitre 3, Section 1.1). La présence du condensateur ne permet pas d'appliquer directement la loi d'Ohm à ce circuit. En effet, aux bornes d'un condensateur, l'intensité dépend de la *dérivée* de la tension par rapport au temps : $I = C \frac{dU}{dt}$. D'après la *loi des nœuds*, l'intensité aux bornes du circuit (I) est la somme des intensités aux bornes de la résistance (I_R) et du condensateur (I_C). On a ainsi $I = I_R + I_C = \frac{U}{R} + C \frac{dU}{dt}$. L'intensité n'est donc pas proportionnelle à la tension dans ce cas. En revanche, il existe bien une relation de proportionnalité entre la charge du condensateur (Q) et la tension à ses bornes (U) avec la capacité du condensateur (C) comme coefficient de proportionnalité : $Q = C \times U$. Dans notre cadre de modélisation, on fait l'analogie entre la charge du condensateur et les ions qui s'accumulent localement à la surface de la membrane. On a ainsi une relation de proportionnalité entre notre notion abstraite de charge électrique et la différence de potentiel classiquement mesurée de part et d'autre de la membrane.

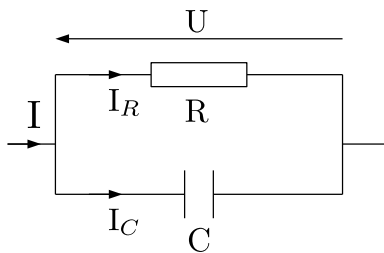


FIGURE 4.1 – Membrane représentée comme un circuit électrique simplifié. La résistance (R) représente les canaux ioniques et le condensateur (C) les propriétés capacitives de la membrane.

Naturellement, les paramètres de notre cadre de modélisation seront fixés sur la base des propriétés biophysiques des neurones biologiques. Néanmoins, notre objectif n'est pas de reproduire les théories biophysiques existantes pour les deux raisons suivantes. Tout d'abord, la modélisation biophysique n'est pas la réalité biologique, mais une approximation de cette dernière. Ensuite, notre objectif n'est pas de décrire en détail les mécanismes biophysiques,

car nous adoptons un point de vue beaucoup plus abstrait en vue d'un passage à l'échelle.

4.1 Description statique d'un neurone

Comme il est d'usage en modélisation formelle de système dynamique, cette section donne une description « statique » de notre modèle de neurone (par opposition à la définition de la dynamique du système). Pour des raisons pédagogiques, on donne dans le même temps une intuition de la dynamique, car les paramètres statiques sont largement motivés par la dynamique. Ces paramètres prendront tout leur sens dans la section d'après qui formalise la dynamique.

De façon assez naturelle par rapport à sa structure (Chapitre 1, Section 1.1), nous modélisons un neurone par un arbre dont la racine est appelée le soma, les feuilles sont appelées des synapses et les autres nœuds internes sont appelés des compartiments (Figure 4.2).

Par rapport à la structure du neurone biologique présentée dans le chapitre 1, nous ignorons ici l'axone. Il est généralement admis en première approximation que la transmission *via* l'axone n'altère pas l'information. Aussi, on néglige le temps de transmission dans l'axone que l'on considère identique pour toutes ses connexions, car la vitesse de propagation est beaucoup plus importante que dans les dendrites. Ce temps pourra facilement être considéré lorsque les neurones seront mis en réseaux.

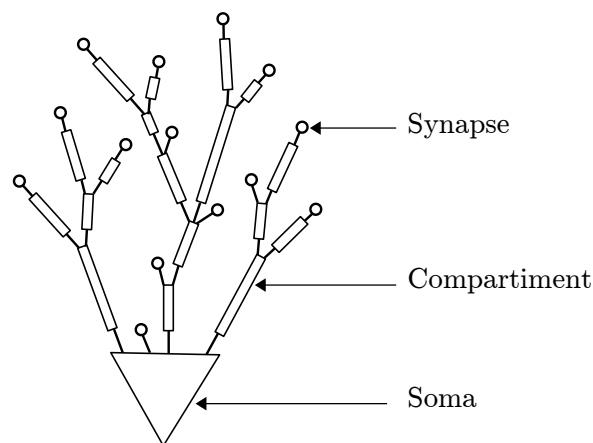


FIGURE 4.2 – Représentation schématique de notre modèle de neurone.

Dans cette section nous allons définir plus précisément chacun des composants d'un neurone puis nous définirons son état à un temps donné.

Nous nous placerons sous l'hypothèse majeure suivante : nous considérons que la propagation des charges dans les dendrites est uniquement *passive* et *unidirectionnelle* (Chapitre 1, Section 1.1).

4.1.1 Les synapses

Les synapses sont les ports d'entrée d'un neurone. Chaque synapse est caractérisée par une *fonction synaptique* (σ), qui modélise la cinétique des potentiels post-synaptiques générés par les spikes reçus (Définition 2 et Figure 4.3 ci-après). Cette cinétique est propre à chaque synapse (Définition 3 ci-après). Ici, nous raisonnons en matière de charge électrique locale et non en valeur de potentiel. Cependant, comme expliqué dans l'introduction de ce chapitre, ces deux grandeurs sont directement liées.

Prenons un exemple simple et intuitif de synapse excitatrice (Figure 4.3a). Au repos ($t \leq 0$), sa fonction synaptique est nulle. En réponse à un potentiel d'action (à $t = 0$), des canaux ioniques s'ouvrent à la membrane post-synaptique. Il se produit alors un courant transmembranaire entrant de charges positives. Ces charges s'accumulent localement sur la face interne de la membrane, de par ses propriétés capacitives. La fonction synaptique augmente alors progressivement jusqu'à atteindre une valeur maximale (w), après un certain temps ($\hat{\tau}$). Puis, à la fermeture des canaux, les charges quittent cette localité pour se déplacer dans le milieu intracellulaire, le long des dendrites. La fonction synaptique décroît ainsi jusqu'à redevenir nulle après un certain temps ($\check{\tau}$).

On caractérise une synapse inhibitrice par une fonction synaptique négative, une synapse inhibitrice ayant l'effet opposé d'une synapse excitatrice.

Définition 2. [Fonction synaptique]

Une fonction synaptique $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ est une fonction continue satisfaisant les conditions suivantes :

- Il existe $\hat{\tau}_\sigma \in \mathbb{R}^{*+}$ et $\check{\tau}_\sigma \in \mathbb{R}^{*+}$ tels que $|\sigma|$ est strictement croissante sur l'intervalle $[0, \hat{\tau}_\sigma]$ et strictement décroissante sur l'intervalle $[\hat{\tau}_\sigma, \hat{\tau}_\sigma + \check{\tau}_\sigma]$;
- Pour tout $t \leq 0$ et pour tout $t \geq \hat{\tau}_\sigma + \check{\tau}_\sigma$, $\sigma(t) = 0$.

Cas particulier affine (Figure 4.3a) : soit $w_\sigma = \sigma(\hat{\tau}_\sigma)$, σ est la fonction synaptique affine sur les intervalles $[0, \hat{\tau}_\sigma]$ et $[\hat{\tau}_\sigma, \hat{\tau}_\sigma + \check{\tau}_\sigma]$ telle que :

- $\forall t \in [0, \hat{\tau}_\sigma], \sigma(t) = \frac{w_\sigma}{\hat{\tau}_\sigma} t$;
- $\forall t \in [\hat{\tau}_\sigma, \hat{\tau}_\sigma + \check{\tau}_\sigma], \sigma(t) = w_\sigma - \frac{w_\sigma}{\check{\tau}_\sigma} (t - \hat{\tau}_\sigma)$.

Définition 3. [Synapse]

Une synapse est un symbole s muni d'une fonction synaptique σ . On note $w_s = \sigma(\hat{\tau}_\sigma)$, $\hat{\tau}_s = \hat{\tau}_\sigma$, $\check{\tau}_s = \check{\tau}_\sigma$ et on appelle :

- w_s le poids de s ;
- $\hat{\tau}_s$ le temps de montée de s ;
- $\check{\tau}_s$ le temps de descente de s .

Si $w_s > 0$, s est dite excitatrice sinon elle est dite inhibitrice.

Remarque: On note que w_s est toujours non nul. Si $w_s > 0$ alors l'image de σ est l'intervalle positif $[0, w_s]$ sinon, l'image de σ est l'intervalle négatif $[w_s, 0]$.

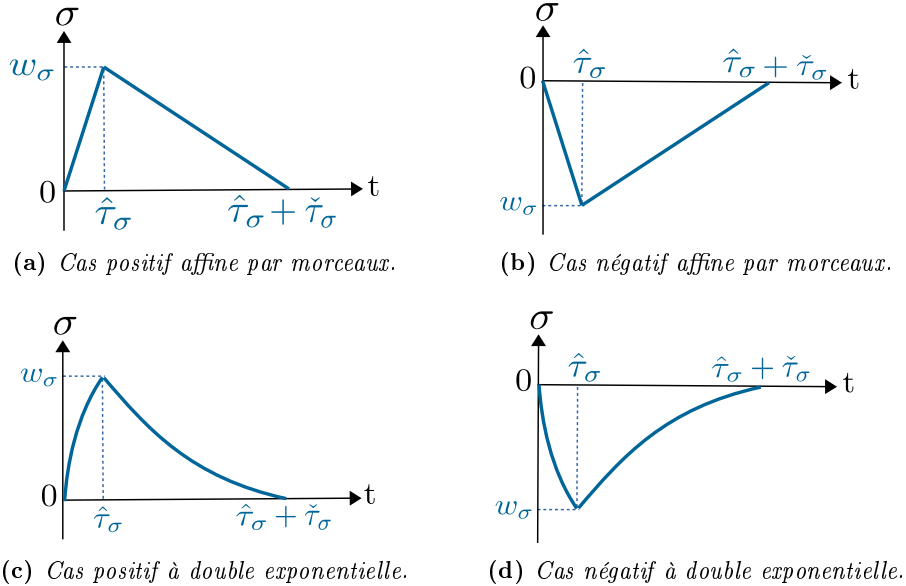


FIGURE 4.3 – Exemples de fonctions synaptiques. Les fonctions synaptiques positives caractérisent des synapses excitatrices alors que les fonctions synaptiques négatives caractérisent des synapses inhibitrices.

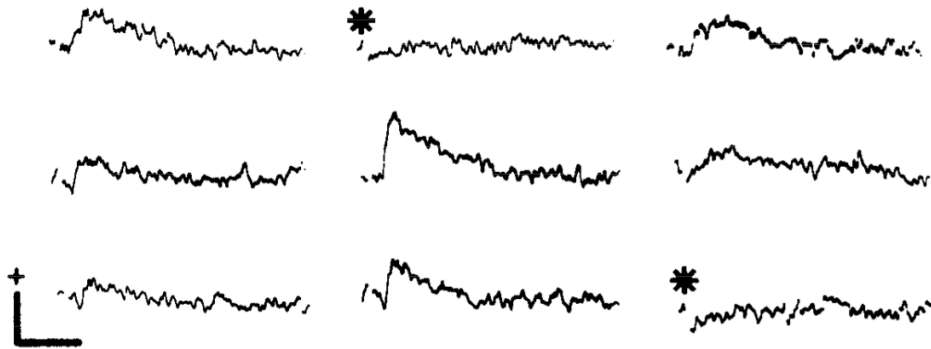


FIGURE 4.4 – Série de neuf enregistrements intracellulaires montrant la variabilité des potentiels post-synaptiques. On note deux échecs de réponses signalés par un astérisque. L'échelle de calibration représente 1mV/10ms. Cette figure montre bien à quel point toute modélisation mathématique est une approximation. **Figure extraite de [McNaughton et al., 1981].**

On présente dans la figure 4.3 des cas particuliers de fonctions synaptiques. En se basant sur les modèles biophysiques existants le cas exponentiel semble plus crédible (Chapitre 3, Section 3.1). Or, l'expérimentation montre que la réponse aux potentiels d'action présente une grande variabilité et ne suit que très approximativement les prédictions biophysiques (Figure 4.4). On ne restreint donc pas la forme des fonctions synaptiques et on s'autorise même le cas particulier d'une approximation linéaire. On retrouve également ce type d'approximation dans les travaux de Maass [Maass, 1999].

L'approximation linéaire n'influe pas sur les résultats théoriques présentés dans cette thèse, mais présente des avantages pour des applications pratiques. Par exemple, nous verrons dans la suite du manuscrit (Partie 6.4) que l'approximation linéaire facilite la discrétisation du temps, ce qui permet notamment de faire des preuves par model checking. Notons que w_σ est un paramètre fixe à ce stade-là, limitant les capacités d'apprentissage du neurone (voir Chapitre 9 pour s'affranchir de cette hypothèse).

4.1.2 Les compartiments

Comme décrit dans les Chapitres 1 et 3, il y a une atténuation du signal au fur et à mesure de la propagation dans les dendrites, due à une fuite de charges à travers la membrane. Cette atténuation dépend notamment de la distance parcourue, du diamètre de la dendrite et de la composition de la membrane.

Ce qui nous intéresse dans ce travail est la fonction d'entrée/sortie du neurone. Par conséquent, nous prenons le parti de l'abstraction au détriment d'une description détaillée de ce qui se passe en chaque point du neurone. Nous définissons un *compartiment* comme une « petite » portion de dendrite que nous choisissons de caractériser par uniquement deux paramètres : le *temps de traversée* (Δ) du compartiment et l'*atténuation* (α) observée à l'issue de celui-ci. À partir de la charge à l'entrée d'un compartiment à un temps donné, il nous sera donc possible de calculer la charge à sa sortie après le temps de traversée. Formellement :

Définition 4. [Compartiment]

Un compartiment est un symbole c muni d'un couple $(\alpha_c, \Delta_c) \in]0, 1] \times \mathbb{R}^{+}$.*

On appelle :

- α_c l'atténuation à l'issue de c ;
- Δ_c le temps de traversée de c .

On s'autorisera la notation $c = (\alpha_c, \Delta_c)$ lorsqu'elle n'induit pas d'ambiguïté.

Dans un compartiment, le temps de propagation du signal est donc constant et l'atténuation se fait de la même manière, quelle que soit la charge.

4.1.3 Le soma

Nous modélisons le soma de manière similaire au classique modèle Leaky Integrate-and-Fire (Chapitre 3, Section 3.1). Par abus de langage, dans notre cadre de modélisation, on appelle *potentiel* (p), la charge électrique au soma.

Le soma accumule au cours du temps toutes les charges provenant des dendrites. Dans le même temps, le potentiel résultant subit continuellement une fuite qui représente, comme dans les compartiments, la fuite de charges à travers la membrane.

Lorsque le potentiel au soma atteint un certain seuil, un spike est émis en sortie et le potentiel est alors immédiatement mis à jour. À ce moment même, commence la *période réfractaire absolue* durant laquelle le neurone ne peut émettre aucun spike en sortie même si le potentiel est supérieur au seuil. Cette période est modélisée par un seuil infini et donc inatteignable. Le seuil retourne ensuite progressivement vers sa valeur normale ce qui modélise la *période réfractaire relative* (Figure 4.5). Cette même approche est utilisée par Maass [Maass, 1999].

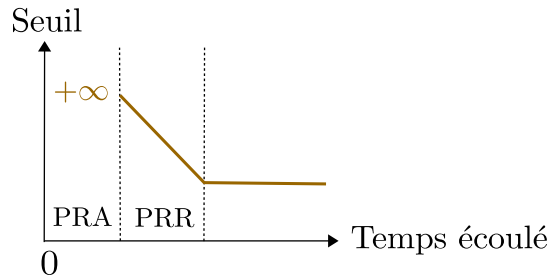


FIGURE 4.5 – Évolution du seuil en fonction du temps écoulé depuis l'émission du dernier spike. Le seuil est infini en période réfractaire absolue (PRA) et augmenté en période réfractaire relative (PRR), par rapport au seuil normal (partie droite de la courbe).

Nous définissons un soma par deux paramètres : un facteur de *fuite* (γ) et une *fonction de borne* (β). La fonction de borne représente l'évolution du seuil au cours du temps écoulé depuis l'émission du dernier spike (e), en prenant en compte les périodes réfractaires (Définition 5 ci-après et Figure 4.6). En raison de l'émission des spikes, certaines valeurs de potentiel p ne sont pas atteignables pour un temps e donné. Par exemple, plaçons-nous hors période réfractaire et prenons une valeur x largement supérieure au seuil d'activation : p n'atteindra jamais la valeur x . En effet, avant d'atteindre une telle valeur, p aura atteint le seuil et un spike aura forcément été émis. À ce même moment, p aura été mis à jour et e aura été réinitialisé. On appelle *nominales* les valeurs atteignables du couple (e, p) (Figure 4.6).

Définition 5. [Fonction de borne]

Une fonction de borne β est une fonction $\beta : \mathbb{R}^+ \cup \{+\infty\} \rightarrow \mathbb{R}^{*+} \cup \{+\infty\}$ satisfaisant les conditions suivantes :

- Il existe $e_\beta \in \mathbb{R}^{*+}$ tel que $\forall e \in [0, e_\beta[, \beta(e) = +\infty$;
- β est continue et décroissante sur l'intervalle $[e_\beta, +\infty]$ et on note $p_\beta = \beta(+\infty) \in [1, +\infty[$;
- Il existe $\hat{p}_\beta \in \mathbb{R}^{*+}$ tel que $\beta(e_\beta) = p_\beta + \hat{p}_\beta \neq +\infty$.

Tous les couples (e, p) satisfaisant la condition $p \leq \beta(e)$ sont dits *nominaux* et on note $Nominal(\beta)$ l'ensemble de tous les couples nominaux. Enfin, β est dite *normale* si et seulement si $p_\beta = 1$.

Remarque: β doit être définie entre 0 et e_β pour faciliter la gestion des couples (e, p) , d'où l'ajout de $+\infty$ à l'ensemble d'arrivée de β . De plus, on impose la contrainte $p_\beta \geq 1$ pour simplifier les définitions des Chapitres 5 et 6 qui permettent d'établir une relation de pré-ordre entre les neurones abstraits. Intuitivement, plus p_β sera proche de 1 et plus le neurone sera considéré minimal. On dira qu'un neurone est normalisé si $p_\beta = 1$. Nous voyons bien ici que p est une valeur abstraite, comme la charge électrique partout ailleurs dans le neurone.

Deux cas particuliers (Figure 4.6) :

- (a) β est une fonction de borne pour laquelle il existe $\hat{p}_\beta, \hat{e}_\beta \in]0, +\infty[$ tel que :
- quel que soit $e \in [e_\beta, e_\beta + \hat{e}_\beta]$: $\beta(e) = p_\beta + \hat{p}_\beta \frac{e_\beta + \hat{e}_\beta - e}{\hat{e}_\beta}$,
 - quel que soit $e \in [e_\beta + \hat{e}_\beta, +\infty]$: $\beta(e) = p_\beta$.
- (b) β est une fonction de borne pour laquelle il existe $\hat{p}_\beta \in]0, +\infty[$ tel que : quelque soit $e \in [e_\beta, +\infty]$, $\beta(e) = p_\beta + (\hat{p}_\beta \cdot \exp(-k \cdot (e - e_\beta)))$ où $k \in \mathbb{R}^{*+}$.

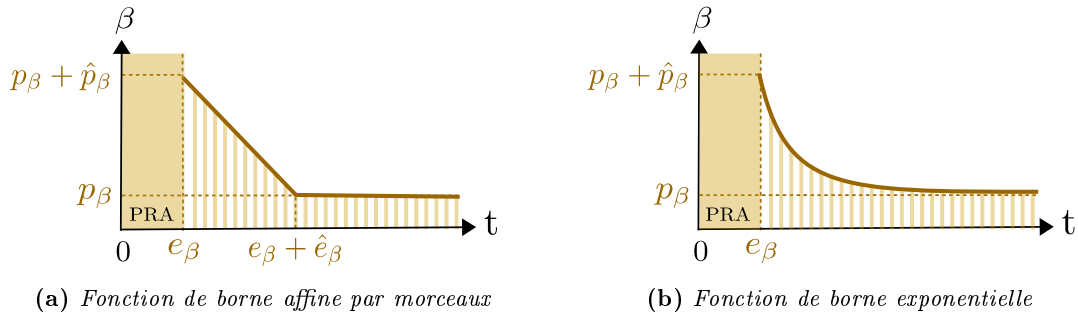


FIGURE 4.6 – Exemples de fonctions de borne. Les valeurs nominales du couple (e, p) correspondent aux valeurs de p inférieures à la fonction de borne. Ces valeurs sont représentées par l'union de toutes les valeurs de p pendant la période réfractaire (PRA) et de la zone hachurée.

Définition 6. [Soma]

Un soma est un symbole ∇ muni d'une fonction de borne β et d'un facteur de fuite $\gamma \in \mathbb{R}^{*+}$. On s'autorise la notation $\nabla = (\beta, \gamma)$ et en se conformant aux notations de la Définition 5, on appelle :

- e_β la période réfractaire absolue de ∇ ;
- p_β le seuil d'activation de ∇ ;
- \hat{p}_β , l'augmentation de seuil par rapport à p_β ;
- et lorsque β est affine par morceaux, \hat{e}_β la période réfractaire relative de ∇ .

De la même manière que pour les fonctions synaptiques, la forme de la fonction de borne n'influe pas sur nos résultats théoriques. En revanche, pour des cas pratiques tels que pour faire des preuves par model checking ou de la simulation, le cas particulier affine par morceaux est intéressant (comme on le verra dans la partie 6.4). On présente également le cas exponentiel sur la Figure, 4.6 car c'est un cas pertinent du point de vue de la biologie (Chapitre 1, Section 1.1) [Misawa et al., 2004] [Markham and Zakon, 2014] [Tuncer et al., 2017].

Effectivement, au sortir de la période réfractaire absolue, le seuil à atteindre pour déclencher l'émission d'un spike est très élevé, puis il diminue rapidement, car les canaux sodiques inactivés par l'émission du précédent spike redeviennent majoritairement actifs. Dans le même temps, la perméabilité des canaux potassiques retourne à son niveau de repos et le seuil retrouve alors progressivement sa valeur habituelle. L'effet réfractaire décroît donc théoriquement de manière exponentielle en fonction du temps écoulé depuis l'émission du dernier spike, avec une constante de temps de l'ordre de la milliseconde [Aihara et al., 1990] [Nedjah et al., 2011] [Platkiewicz and Brette, 2010].

Maintenant que nous avons défini les différents composants d'un neurone, nous allons définir formellement ce qu'est un neurone dans son ensemble.

4.1.4 Le neurone complet

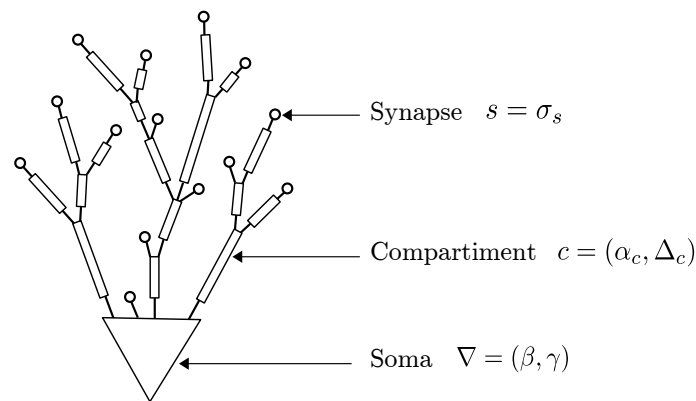


FIGURE 4.7 – Représentation schématique de notre modèle de neurone.

En informatique, un arbre enraciné est un graphe sans cycle dans lequel on désigne un nœud racine. Ceci oriente *de facto* le graphe. Notre modèle de neurone est un arbre dont la racine représente le soma et les sous-arbres représentent les dendrites dont les nœuds sont soit des synapses (aux feuilles), soit des compartiments (Figure 4.7). Formellement :

Définition 7. [Neurone]

Un neurone N est un arbre enraciné non trivial (c'est-à-dire non réduit à sa racine) et étiqueté tel que :

- la racine est étiquetée par un soma ;
- tout autre nœud interne est étiqueté par un compartiment ;
- toute feuille est étiquetée par une synapse ;
- les symboles utilisés doivent être différents tout au long de l'arbre (chaque étiquette doit être utilisée de manière unique).

On note $Sy(N)$ l'ensemble des synapses de N , $Co(N)$ l'ensemble des compartiments de N et on note $Nodes(N)$ l'ensemble $Sy(N) \cup Co(N) \cup \{\nabla\}$. De plus, on appelle forêt dendritique de N , l'ensemble des sous-arbres enracinés en ∇ .

Enfin, on note \mathcal{N} l'ensemble de tous les neurones.

Un neurone est un système orienté, qui propage les signaux des feuilles (synapses) vers la racine (soma). De manière inhabituelle pour les arbres, un neurone est ainsi représenté « à l’endroit », c’est-à-dire que les feuilles sont en haut et la racine en bas. Pour éviter toute confusion, nous utilisons donc le terme *prédécesseur* pour désigner le fils d’un nœud (Figure 4.8) :

Terminologie 1. [Prédécesseur]

Étant donné un neurone N , on appelle *prédécesseur* d’un nœud $x \in \text{Nodes}(N)$, tout nœud fils de x dans l’arbre sous-jacent de N . On note $\text{Pred}_N(x)$ l’ensemble des prédécesseurs de x dans N . De plus, on note $\text{SyPred}_N(x)$ l’ensemble des synapses appartenant à l’ensemble $\text{Pred}_N(x)$ et $\text{CoPred}_N(x)$ l’ensemble des compartiments appartenant à ce même ensemble.

Remarque: $\text{Pred}_N(x)$ est l’union disjointe $\text{CoPred}_N(x) \amalg \text{SyPred}_N(x)$.

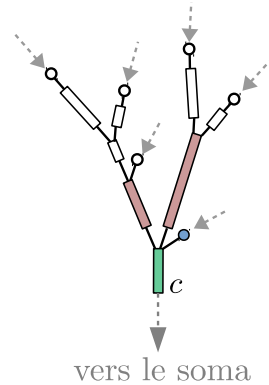


FIGURE 4.8 – Prédécesseurs d’un compartiment dans un neurone N . Soit c le compartiment vert, les prédécesseurs de c dans N ($\text{Pred}_N(c)$) sont les deux compartiments roses et la synapse bleue. Les deux compartiments roses forment l’ensemble $\text{CoPred}_N(c)$ et la synapse bleue constitue à elle seule l’ensemble $\text{SyPred}_N(c)$. Les flèches grises représentent le sens de propagation des signaux.

Une autre possibilité aurait été de représenter les compartiments par des arcs et non des nœuds (Figure 4.9).

Cette définition alternative présente deux inconvénients principaux :

- il faudrait représenter explicitement les points d’embranchement par des nœuds ; par exemple, un compartiment (arc) ne pourrait pas être directement connecté à deux compartiments prédécesseurs (arcs), il faudrait impérativement que le compartiment en question soit connecté à un nœud, lui-même connecté aux compartiments prédécesseurs.
- il faudrait autoriser des *compartiments nuls*, c’est à dire des compartiments dont le temps de traversée serait nul et le facteur d’atténuation égal à 1 ; par exemple, une synapse (nœud) que l’on souhaiterait « collée » au soma devrait être connectée à ce dernier par un compartiment (arc) nul.

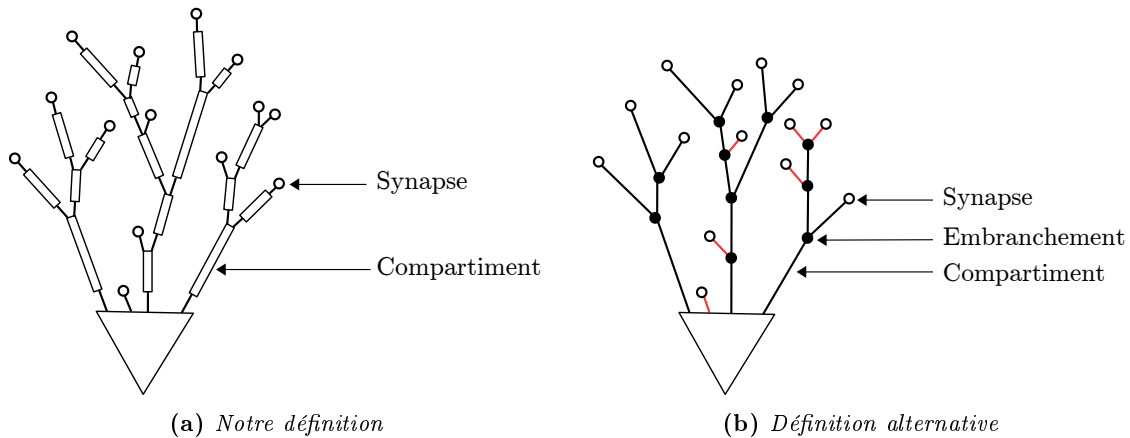


FIGURE 4.9 – Définition alternative d'un neurone. Dans notre définition, les compartiments comme les synapses sont représentés par les nœuds du graphe. Il existe une définition alternative qui représenterait les compartiments comme des arcs. Les arcs rouges sont des compartiments nuls.

Dans cette définition alternative, les points d'embranchement connecteraient un compartiment soit à des compartiments prédécesseurs non nuls, soit à des compartiments prédécesseurs nuls liés eux-mêmes à des synapses. Les points d'embranchements, tout comme les compartiments nuls, n'influencent pas le comportement du neurone, mais alourdissent les définitions, ce qui est en défaveur de cette version alternative.

Dans cette version alternative, ainsi que dans notre définition, toutes les feuilles d'un neurone sont des synapses. En effet, des compartiments feuilles, qui ne seraient pas alimentés par des synapses, n'auraient pas d'entrée et donc pas de raison d'être. Dans le cas particulier où les points d'embranchement ne seraient pas autorisés à lier un compartiment à un compartiment prédécesseur nul (Figure 4.10), les synapses pourraient alors également être des nœuds internes (et pas uniquement des feuilles).

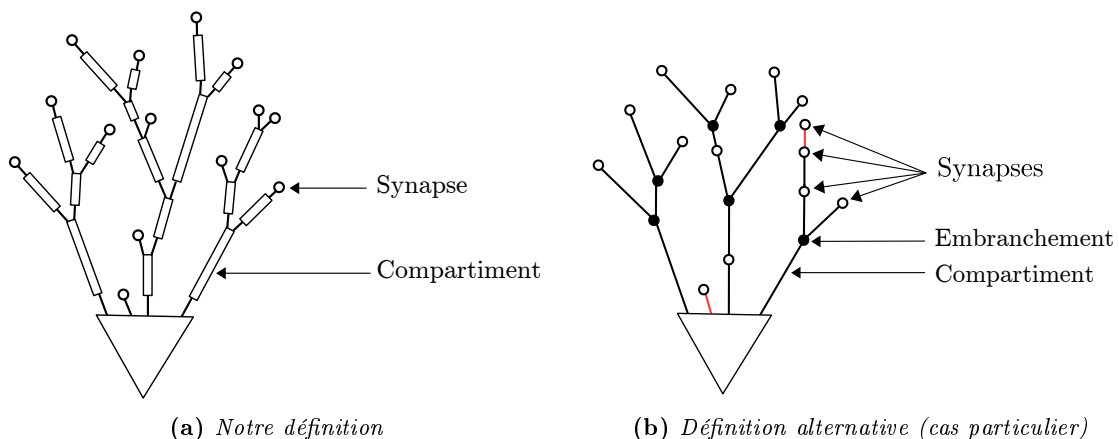


FIGURE 4.10 – Définition alternative d'un neurone : cas particulier pour lequel un point d'embranchement ne peut pas lier un compartiment à un compartiment prédécesseur nul. Dans ce cas, les synapses ne sont pas forcément des feuilles. Les arcs rouges sont des compartiments nuls.

Avec ce cas particulier, la représentation serait allégée et plus cohérente avec la biologie, dans

le sens où les synapses ne sont pas nécessairement localisées aux extrémités des dendrites. En revanche, cette version complexifierait d'autant plus nos définitions. Nous verrons dans la sous-section suivante que l'état d'un compartiment dépend notamment de l'état de ses synapses prédécesseuses. Dans cette version alternative, l'état d'un compartiment pourrait également dépendre de l'état d'autres synapses. Prenons un exemple simple (Figure 4.11) : dans notre définition, le compartiment vert a deux prédécesseurs (les deux synapses bleues) alors qu'il n'en aurait qu'un dans la version alternative (la synapse bleue). Le compartiment rouge étant nul, la synapse rouge contribue autant à l'état du compartiment vert que la synapse bleue. Avec la définition alternative, il faudrait donc étendre la notion de prédécesseur pour inclure cette synapse rouge.

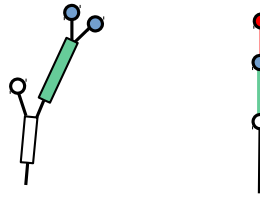


FIGURE 4.11 – Zoom sur une « portion » de neurone. À gauche, notre définition et à droite, la définition alternative. Dans notre définition, le compartiment vert a deux synapses prédécesseuses alors que dans la définition alternative, le compartiment vert a une unique synapse prédécesseuse (la synapse bleue). En revanche, l'arc rouge étant un compartiment nul, la synapse rouge contribue tout autant que la synapse bleue à l'état du compartiment vert.

Nous avons donc choisi de représenter les synapses ainsi que les compartiments par des nœuds du graphe, car cette version simplifie les définitions. De plus, ce choix homogénéise les notations puisque « tout est nœud » dans un neurone.

Dans la sous-section suivante, nous définissons l'état d'un neurone. Nous construirons par la suite la dynamique comme étant une « succession » d'états, dépendant d'un temps *continu*.

4.1.5 État d'un neurone

Nous définissons l'état η d'un neurone N (à un temps donné) comme la valeur de la charge électrique en tous ses points. On distingue l'état de l'ensemble des synapses $\{\eta_s\}_{s \in Sy(N)}$, l'état de l'ensemble des compartiments $\{\eta_c\}_{c \in Co(N)}$ et l'état du soma η_∇ .

État des synapses

L'état d'une synapse s (η_s) (à un temps donné) est un unique nombre réel qui représente la valeur de la charge électrique à la synapse à cet instant. Pour rappel, en biologie, une synapse désigne la jonction entre deux neurones. On admet donc qu'une synapse est ponctuelle ce qui justifie cette définition de l'état.

État des compartiments

Contrairement aux synapses, les compartiments ne sont pas ponctuels. De manière intuitive, l'état d'un compartiment c (η_c) (à un temps donné) représente la valeur de la charge électrique en chaque point de c à cet instant.

Nous avons défini un compartiment c en utilisant deux paramètres : son délai de traversée Δ_c et l'atténuation qu'il entraîne α_c . Le paramètre α_c permet de calculer, à partir de la charge en entrée de c à un temps donné, la charge en sortie de c après un intervalle de temps Δ_c : la charge initiale est simplement multipliée par α_c . La multiplication par un réel inférieur à 1 est plus pertinente du point de vue biologique que par exemple une soustraction de quantité fixe. En effet, au cours de la propagation dans les dendrites, la quantité de charges qui fuit à travers les canaux transmembranaires n'est pas fixe : plus il y a de charges et plus la fuite est importante (en condition de non-saturation). Sur ce point, notre théorie va dans le sens de la théorie biophysique en conditions d'état stable (Chapitre 3, Section 3.1).

Bien que Δ_c soit une durée et non une distance, il est relativement facile de passer d'une grandeur à l'autre. Puisque l'état de c est la valeur de la charge en chacun de ses points, si c est de longueur L_c , c'est une fonction $f : [0, L_c] \rightarrow \mathbb{R}$ qui associe à chaque distance d de sa sortie (Figure 4.12), la charge en ce point. Cependant, si on note δ le temps pour parcourir la distance d , il est équivalent de définir une fonction $\eta_c : [0, \Delta_c] \rightarrow \mathbb{R}$ qui, à toute durée $\delta < \Delta_c$, associe une valeur de charge locale. Nous avons ainsi choisi de définir l'état de c par une fonction temporelle, ce qui simplifie nos définitions.

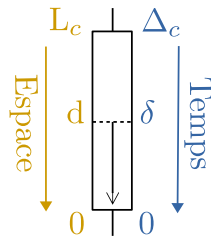


FIGURE 4.12 – *Abstraction du temps et de l'espace.* On définit l'état d'un compartiment c par un segment qui associe une valeur de charge locale à chaque temps $\delta \in [0, \Delta_c]$. Considérer δ (qui est le temps pour aller d'un point donné à la sortie de c) est équivalent à considérer d (qui est la distance entre ces deux points).

Le paramètre Δ_c est donc une durée qui représente aussi une distance de manière abstraite. On définit de cette façon η_c à un temps donné par un segment (Chapitre 2, Section 2.1, Définition 1) portant les valeurs de la charge à la sortie de c à partir de ce temps donné et pour une durée de Δ_c . Dès lors, $\eta_c(0)$ est la valeur de la charge en sortie de c au temps courant et de manière similaire, $\eta_c(\delta)$ est la valeur de la charge en sortie de c après un délai δ , c'est à dire « dans le futur » par rapport à l'instant courant. La valeur $\eta_c(\Delta_c)$ correspond donc à la valeur de la charge en entrée de c au temps courant, multipliée par le facteur d'atténuation α_c .

Pour assurer la cohérence de l'état, il faut respecter la condition de continuité de la Définition 8 donnée plus bas. Intuitivement, il s'agit d'une condition de conservation : par exemple, les charges sortant d'un compartiment doivent être préservées à l'entrée du compartiment suivant. Plus précisément, la charge en entrée d'un compartiment c doit être égale à la somme des charges provenant des nœuds prédécesseurs de c (compartiments et synapses).

État du soma

On considère le soma ponctuel, car on admet que les charges s'y accumulent de manière homogène. Ainsi, on définit l'état du soma à un temps donné par un couple de nombres réels (e, p) .

- p représente le potentiel du soma, c'est-à-dire notre variable d'intérêt ;
- e représente le temps écoulé depuis l'émission du dernier spike, permettant de gérer les périodes réfractaires.

État d'un neurone complet

La définition ci-après définit l'état d'un neurone qui inclut l'état des synapses, l'état des compartiments et l'état du soma.

Définition 8. [État d'un neurone]

Étant donné un neurone N , un état de N est un triplet $\eta = (\{\eta_s\}_{s \in Sy(N)}, \{\eta_c\}_{c \in Co(N)}, \eta_\nabla)$ tel que :

- $\{\eta_s\}_{s \in Sy(N)}$ est une famille de nombres réels indexée par $Sy(N)$ où chaque élément $\eta_s \in \mathbb{R}$ est appelé l'état de la synapse s .
- $\{\eta_c\}_{c \in Co(N)}$ est une famille de segments indexée par $Co(N)$ où chaque élément est de la forme $\eta_c : [0, \Delta_c] \rightarrow \mathbb{R}$ et vérifie la condition de continuité suivante :

$$\eta_c(\Delta_c) = \left(\sum_{c' \in CoPred_N(c)} \eta_{c'}(0) + \sum_{s \in SyPred_N(c)} \eta_s \right) \times \alpha_c$$

Chaque segment η_c est appelé l'état du compartiment c .

- η_∇ est un couple $(e, p) \in Nominal(\beta)$ où β est la fonction de borne du soma de N . η_∇ est appelé l'état du soma de N , e est appelé le temps passé depuis l'émission du dernier spike et p est appelé le potentiel de soma.

On note ζ_N l'ensemble de tous les états possibles de N .

On définit également l'état dendritique ∂ qui regroupe l'état des synapses et l'état des compartiments, permettant de se passer du soma dans de nombreuses définitions :

Définition 9. [État dendritique d'un neurone]

Étant donné un neurone N et un état η de N , la restriction de cet état à la forêt dendritique de N , $\partial = (\{\eta_s\}_{s \in Sy(N)}, \{\eta_c\}_{c \in Co(N)})$, est appelée un état dendritique de N . On note \mathcal{D}_N l'ensemble de tous les états dendritiques possibles de N .

Étant donnée la description statique présentée dans cette section, nous définissons dans la section suivante la description dynamique du neurone comme une « succession » d'états au cours du temps, plus précisément comme une fonction qui associe à chaque temps $t \in \mathbb{R}^+$ un état de N .

4.2 Description de la dynamique d'un neurone

La description dynamique de notre cadre de modélisation formalise les processus qui permettent à un neurone d'intégrer des signaux d'entrée pour produire un unique signal de sortie.

4.2.1 Signaux discrets

Les signaux reçus aux synapses sont des segments (Chapitre 2, Section 2.1, Définition 1) infinis ayant pour valeur 1 aux temps des spikes et 0 le reste du temps. On suppose ainsi que le spike est un événement ponctuel (de durée nulle), ayant une amplitude constante que l'on peut normaliser à 1. Il s'agit d'une représentation habituelle en neurosciences computationnelles. Dans le contexte informatique, cette représentation booléenne est préférable à l'emploi de diracs qui impliquerait de considérer l'infini aux temps des spikes.

Il existe un intervalle minimal entre deux spikes, dû à la période réfractaire du neurone dont provient le signal. Ainsi, sur un intervalle donné, le nombre de spikes maximal est fini. De plus, le *support* est simplement l'ensemble des temps auxquels il y a des spikes. On appelle *signal élémentaire* une suite de spikes et *signal d'entrée* l'ensemble des suites de spikes reçues par toutes les synapses d'un neurone. Formellement :

Définition 10. [Signal élémentaire]

Un signal élémentaire est un segment $\omega : \mathbb{R} \rightarrow \{0, 1\}$ tel que :

$$\exists r \in \mathbb{R}^{*+}, \forall t \in \mathbb{R}, (\omega(t) = 1 \Rightarrow (\forall t' \in]t, t + r[, \omega(t') = 0))$$

Le support de ω est défini par : $Car(\omega) = \{t \in \mathbb{R} \mid \omega(t) = 1\}$. De plus, un signal élémentaire tel que $Car(\omega)$ est un singleton $\{u\}$ est appelé spike au temps u , noté ω^u .

Remarque: Un signal élémentaire ω peut être fractionné en une somme de spikes séparés par au moins r : $\omega = \sum_{u \in \text{Car}(\omega)} \omega^u$. Cette somme est potentiellement infinie, mais elle est dénombrable et toujours finie sur un intervalle de temps borné, car l'ensemble $\text{Car}(\omega) \cap [t_1, t_2]$ a nécessairement un cardinal inférieur ou égal à $(t_2 - t_1)/r$.

Un signal élémentaire ω est une notion générale. Pour plus de clarté, on notera I_s (pour *Input*) un signal élémentaire reçu à une synapse s et O (pour *Output*) un signal de sortie :

Définition 11. [Signal d'entrée]

Étant donné un neurone N , un signal d'entrée pour N est une famille de signaux élémentaires $I = \{I_s\}_{s \in \text{Sy}(N)}$ indexée par $\text{Sy}(N)$.

Notation 1. [Restriction d'un signal d'entrée à une synapse]

Étant donné un neurone N , un signal d'entrée $I = \{I_s\}_{s \in \text{Sy}(N)}$ pour N et une synapse $s_0 \in \text{Sy}(N)$. On note $I_{|s_0} = \{I'_s\}_{s \in \text{Sy}(N)}$ le signal d'entrée pour N tel que $I'_{s_0} = I_{s_0}$ et $I'_s(t) = 0$ pour toute synapse $s \in \text{Sy}(N) \setminus \{s_0\}$ et quel que soit $t \in \mathbb{R}$.

$I_{|s_0}$ est appelé la restriction de I à la synapse s_0 .

Remarque: Un signal d'entrée peut être fractionné suivant l'ensemble des synapses : $I = \sum_{s \in \text{Sy}(N)} I_{|s}$. Donc $I = \sum_{s \in \text{Sy}(N)} \sum_{u \in \text{Car}(I_s)} I_s^u$ fournit une décomposition de I en spikes élémentaires.

Étant donné un neurone N , un état initial η et un signal d'entrée I , l'objectif de la construction mathématique qui suit est de construire le *signal de sortie* de N , noté O . Le signal de sortie sera donc du même type que le signal d'entrée, ouvrant la possibilité de connecter des neurones en réseau dans lequel les entrées aux synapses d'un neurone sont issues de la sortie d'autres neurones.

4.2.2 Du discret au continu : la dynamique des synapses

Lorsqu'une synapse reçoit un spike, on modélise le potentiel post-synaptique par une *trace* (voir Définition 12 ci-après et Figure 4.13). Cette trace, qui représente plus précisément la charge électrique à l'origine du potentiel post-synaptique, dépend directement de la fonction synaptique σ caractérisant la synapse. Localement, comme décrit dans la section précédente, la valeur absolue de la charge (et donc l'état de la synapse) augmente progressivement après la réception d'un spike puis diminue jusqu'à redevenir nulle. La trace sur une synapse inhibitrice est donc le miroir de la trace sur une synapse excitatrice. Pour des raisons pédagogiques, nous prendrons généralement, dans ce qui suit, l'exemple des synapses positives à fonction synaptique affine par morceaux, mais dans les énoncés, σ reste quelconque.

4.2. DESCRIPTION DE LA DYNAMIQUE D'UN NEURONE

Dans un neurone biologique, un spike en entrée ne suffit généralement pas pour déclencher l'émission d'un spike. Comme mentionné dans le chapitre 1 (Section 1.1), deux phénomènes principaux permettent l'atteinte du seuil : la sommation temporelle et la sommation spatiale. Au niveau de la dynamique des synapses, nous modélisons le phénomène de sommation temporelle en sommant les traces respectives des spikes reçus successivement (voir Figure 4.13). La sommation spatiale sera décrite dans la sous-section suivante.

Définition 12. [Trace d'un signal sur une synapse]

Étant donnée une synapse s d'un neurone N , la trace d'un spike I_s^u sur s au temps u est un segment $\bar{I}_s^u : \mathbb{R} \rightarrow \mathbb{R}$ défini par :

$$\bar{I}_s^u(t) = \sigma_s(t - u)$$

Étant donné un signal d'entrée $I = \{I_s\}_{s \in \text{Sy}(N)}$ pour N , la trace de I_s sur s est définie pour tout $t \in \mathbb{R}$, par le segment $\bar{I}_s(t) = \text{trace}(I_s)(t) = \sum_{u \in \text{Car}(I_s)} \bar{I}_s^u(t)$. Cette somme est bien définie pour tout t car c'est une somme finie de segments en tout point.

Remarque: Étant donnée une synapse s , on note $TC_s = \hat{\tau}_s + \check{\tau}_s$. TC_s représente le temps durant lequel un spike contribue à l'état de la synapse s (Figure 4.13). Pour tout $t \in \mathbb{R}$, on a donc $\bar{I}_s(t) = \text{trace}(I_s|_{[t-TC_s, t]})(t)$.

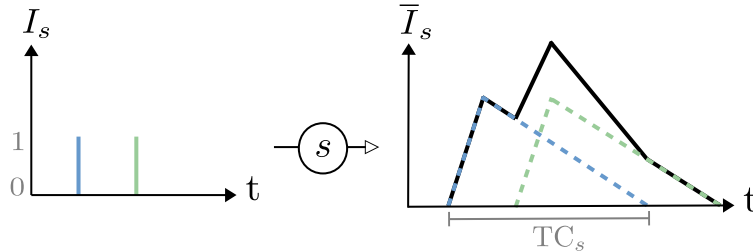


FIGURE 4.13 – Trace d'un signal élémentaire sur une synapse s . Les lignes en pointillés représentent les traces produites individuellement par chaque spike et la ligne pleine représente la trace globale obtenue par sommation. TC_s est le temps de contribution d'un spike à la trace globale sur s . En effet, dans l'exemple du « spike bleu », la trace individuelle n'a plus d'influence sur la trace globale (valeur de \bar{I}_s) après un intervalle de temps TC_s . Pour des raisons pédagogiques, on se limite ici au cas particulier d'une fonction synaptique positive affine par morceaux, mais on retrouve le même phénomène de sommation dans le cas général.

Lorsque des spikes successifs sont reçus au niveau d'une synapse, on fait l'hypothèse que le courant transmembranaire généré à la synapse est constant. Dans ce cas, l'analogie membrane/condensateur largement adoptée en biophysique prédit une évolution exponentielle de la tension et donc de la charge localement [Shepherd, 2004]. Ce type de réponse est également observé expérimentalement par patch clamp en courant imposé, une expérience qui a pour but de mesurer la variation du voltage à la membrane en réponse à l'application d'un courant d'intensité connue (Figure 4.14).

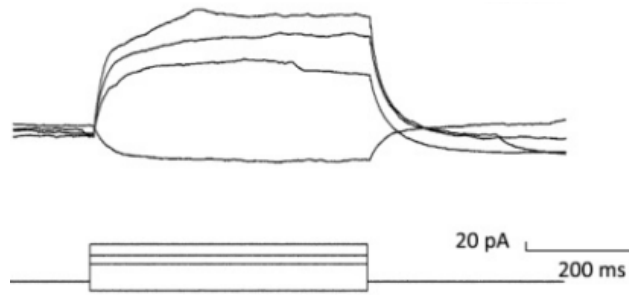


FIGURE 4.14 – Réponses en voltage (en haut) à des expériences représentatives de patch clamp en courant imposé (en bas) sur des neurones spontanément actifs. On constate le comportement exponentiel de la réponse causée par le courant imposé, que le courant soit hyperpolarisant ou dépolarisant. *Figure extraite de [Gunhanlar et al., 2017].*

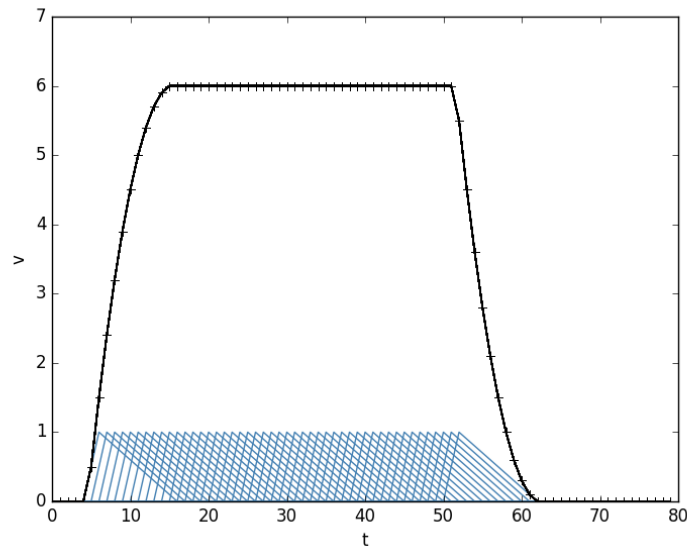


FIGURE 4.15 – Comportement exponentiel de la somme des traces de spikes successifs sur une synapse s à fonction synaptique positive affine par morceaux. Les traces élémentaires sont représentées en bleu et la trace globale obtenue par sommation temporelle des traces élémentaires est représentée en noir. Les paramètres utilisés sont choisis arbitrairement ($w_s = 1$, $\hat{\tau}_s = 2$, $\check{\tau}_s = 10$).

La sommation telle que nous la modélisons, reproduit effectivement un comportement exponentiel global, et ce même lorsque l'on considère l'approximation linéaire des fonctions synaptiques (Figure 4.15). Prenons l'exemple simple d'une synapse s qui reçoit régulièrement des spikes séparés par un temps supérieur au temps de montée $\hat{\tau}_s$ et inférieur à $\hat{\tau}_s + \check{\tau}_s$. Ainsi, la trace d'un spike est sommée à la trace du spike précédent, sur sa pente descendante. Plus il y a de traces successives et plus la trace globale, c'est-à-dire la somme, augmente. Cependant, elle augmente de moins en moins vite, car le nombre de descentes (pentes négatives) augmente. À partir d'un certain moment, la trace globale atteint un plateau, car les descentes compensent les montées. Puis, à l'arrêt du signal, la trace globale diminue progressivement, de moins en moins rapidement, car le nombre de descentes qui contribuent à la trace globale diminue au cours du temps.

4.2. DESCRIPTION DE LA DYNAMIQUE D'UN NEURONE

L'état des synapses, et donc du neurone, dépend des spikes reçus dans le passé. On dit que l'état d'un neurone est *en continuité avec un signal d'entrée* si l'état de chaque synapse est cohérent avec la trace des spikes reçus (Définition 13 ci-après). À un temps donné, un signal d'entrée et sa trace associée sont toujours « centrés » sur zéro, les temps négatifs représentent le passé et similairement les temps positifs représentent le futur. La propriété de continuité est satisfaite si et seulement si, pour toute synapse, son état à un temps donné est égal à la trace du signal d'entrée sur celle-ci au même temps (c'est à dire au temps 0).

Définition 13. [Continuité d'un état avec un signal d'entrée]

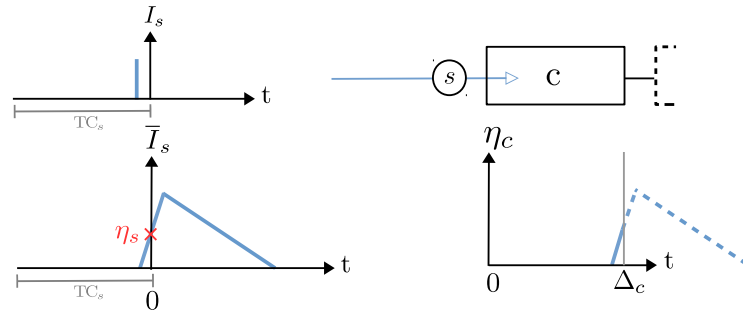
Soient un neurone N , un signal d'entrée $I = \{I_s\}_{s \in Sy(N)}$ et un état $\eta = (\{\eta_s\}_{s \in Sy(N)}, \{\eta_c\}_{c \in Co(N)}, \eta_\nabla)$ de N . L'état η est en continuité avec I si et seulement si pour toute synapse $s \in Sy(N)$,

$$\eta_s = \bar{I}_s(0)$$

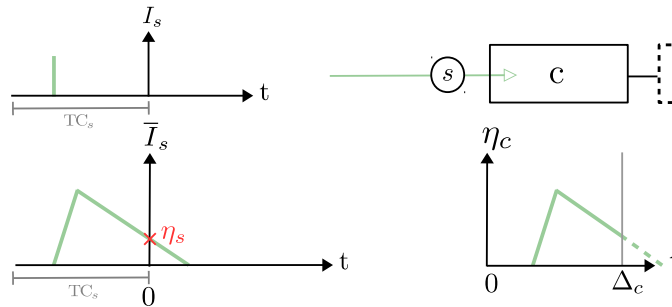
Soit $\partial = (\{\eta_s\}_{s \in Sy(N)}, \{\eta_c\}_{c \in Co(N)})$, la restriction dendritique de η , on définit de manière similaire la continuité de ∂ avec I .

La trace au temps 0 dépend des spikes reçus dans le passé, sur une fenêtre temporelle de durée TC_s propre à chaque synapse (Remarque de la Définition 12). Cependant, notre Définition 13 vérifie la continuité de l'état avec un signal d'entrée seulement au temps courant, en 0 (Figure 4.16), sans tenir compte de la *cohérence* sur l'intervalle $[-TC_s, 0[$. Il s'agit là d'une considération technique qui n'influe pas sur la théorie :

Comme le montre la figure 4.16, la Définition 13 est plus permissive que nécessaire. Elle autorise des « recollements » non conformes entre certains états et certains signaux d'entrée. En effet, pour toute synapse s , tout l'intervalle $[-TC_s, 0]$ du signal d'entrée I_s et donc de la trace \bar{I}_s détermine l'état des compartiments successeurs de s jusqu'à un temps TC_s (Figure 4.16). Il faudrait donc a minima que l'état du compartiment successeur de s soit cohérent avec la trace \bar{I}_s . En effet, la charge à la synapse est instantanée, et la charge qui se trouvait à la synapse à un temps passé se retrouve donc propagée dans les compartiments au temps courant. Ce qui justifie la permissivité de la Définition 13, c'est que tous nos théorèmes restent vrais, mais qu'en revanche assurer la cohérence totale conduirait à une définition beaucoup plus complexe en raison de la question suivante : jusqu'à quel compartiment successeur de s faudrait-il vérifier la cohérence ? Pour une synapse s donnée, si son compartiment successeur c avait un temps de traversée Δ_c supérieur ou égal à TC_s , il suffirait de vérifier la cohérence avec l'état de ce compartiment (Figure 4.16). En revanche, si Δ_c était inférieur à TC_s , il faudrait également vérifier la cohérence avec l'état des compartiments suivants. Ajouter cette contrainte de cohérence n'est donc pas gratuit. Nous verrons dans le chapitre suivant que la propriété de continuité en 0 suffit pour prouver nos théorèmes d'équivalence et nous choisissons donc de ne pas tenir compte de la cohérence sur l'intervalle $[-TC_s, 0[$, ce qui offre des résultats plus généraux.



(a) Exemple d'un signal d'entrée pour lequel l'état de la synapse s est croissant.



(b) Exemple d'un signal d'entrée pour lequel l'état de la synapse s est décroissant.

FIGURE 4.16 – Exemples de signaux I_s ayant des traces (\bar{I}_s) différentes sur une synapse s mais la même valeur en zéro. En accord avec la Définition 13, l'état de s (η_s) est en continuité avec le signal de l'exemple (a) et le signal de l'exemple (b), car $\eta_s = \bar{I}_s(0)$ quelque soit l'exemple. En revanche, l'état du compartiment c (successeur de s) dans l'exemple (a) est cohérent avec le signal de l'exemple (a), mais pas avec le signal de l'exemple (b). De même, l'état de c dans l'exemple (b) est cohérent avec le signal de l'exemple (b), mais pas avec le signal de l'exemple (a).

4.2.3 Des synapses au soma : la dynamique dendritique

Les traces générées aux synapses en réponse aux spikes reçus se propagent ensuite à travers la forêt dendritique compartiment après compartiment, en direction du soma. Alors qu'à la synapse on observe ponctuellement la charge au cours du temps, dans les compartiments, on veut connaître l'évolution de la charge tout au long de son déplacement vers le soma.

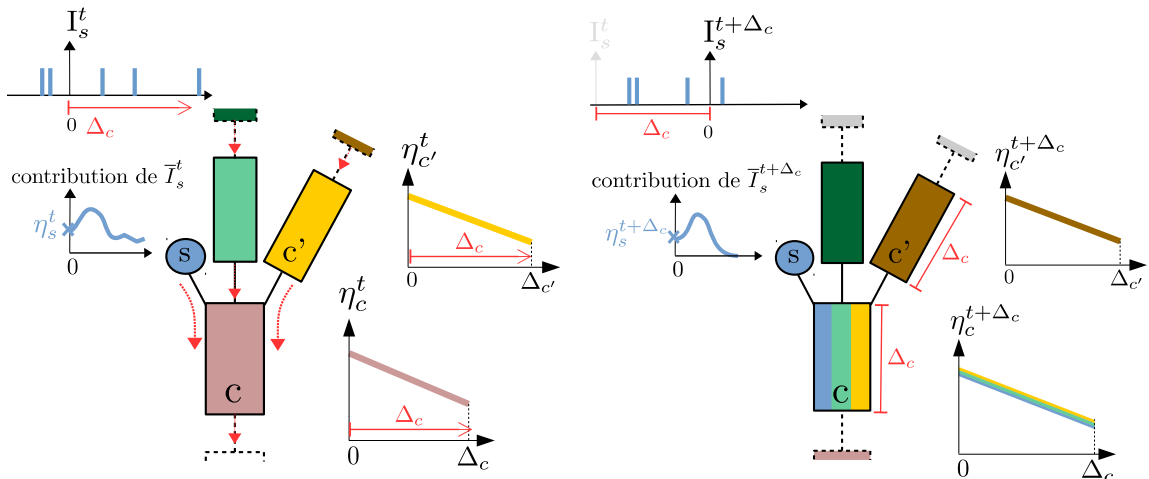
Pour rappel, on suppose ici une propagation passive et unidirectionnelle. En traversant un compartiment c , la charge est multipliée par le facteur d'atténuation α_c qui le caractérise, après un temps de parcours défini Δ_c . Par ailleurs, en accord avec la condition de continuité de la Définition 8 qui modélise le phénomène de sommation spatiale, la charge entrant dans un compartiment est la somme des charges provenant de ses nœuds prédécesseurs (synapses et compartiments). Étant donné ces deux propriétés, il est possible de formaliser la dynamique des dendrites par glissements ou *shifts* successifs (Définition 15 et Théorème 1 ci-après).

Prenons d'abord l'exemple simple d'un neurone où tous les compartiments ont le même

4.2. DESCRIPTION DE LA DYNAMIQUE D'UN NEURONE

temps de traversée Δ_c sur lesquels on applique un tel shift de durée Δ_c (Figure 4.17). C'est-à-dire qu'à partir de l'état des dendrites à un temps t (que l'on note η^t), on calcule l'état des dendrites à $t + \Delta_c$ (que l'on notera donc $\eta^{t+\Delta_c}$).

- Au temps t (Figure 4.17a), l'état η_c^t d'un compartiment c correspond à la charge à la sortie de c entre les temps t et $t + \Delta_c$. $\eta_c^t(0)$ est la charge à la sortie de c au temps t et $\eta_c^t(\Delta_c)$ est la charge à la sortie de c au temps $t + \Delta_c$. De plus, $\eta_c^t(\Delta_c)$ correspond à la charge à l'entrée de c au temps t (atténuée par α_c). Cette charge est elle-même la somme des charges à la sortie des nœuds prédécesseurs de c au même temps.
- Au temps $t + \Delta_c$ (Figure 4.17b), c'est-à-dire après le shift, l'état $\eta_c^{t+\Delta_c}$ d'un compartiment c correspond à la charge à la sortie de c entre les temps $t + \Delta_c$ et $(t + \Delta_c) + \Delta_c$.
 - La charge à la sortie de c à $t + \Delta_c$ (soit $\eta_c^{t+\Delta_c}(0)$) peut directement être calculée à partir de l'état de c au temps t . En effet, $\eta_c^{t+\Delta_c}(0)$ est égal à $\eta_c^t(\Delta_c)$ c'est à dire à la charge à l'entrée de c à t atténuée par α_c .
 - La charge à la sortie de c au temps $(t + \Delta_c) + \Delta_c$ (soit $\eta_c^{t+\Delta_c}(\Delta_c)$) peut être calculée à partir de l'état des nœuds prédécesseurs de c au temps t . En effet, la charge à la sortie de c au temps $(t + \Delta_c) + \Delta_c$ est égale à la charge à l'entrée de c au temps $(t + \Delta_c)$ atténuée par α_c , c'est à dire à la somme des charges à la sortie des nœuds prédécesseurs de c au temps $(t + \Delta_c)$ atténuée par α_c , ce qui est égal pour chaque compartiment prédécesseur c' de c , à la charge à l'entrée de c' au temps t atténuée par α_c ce qui correspond à la valeur de $\eta_{c'}^t(\Delta_c)$.
- En fait, le raisonnement est similaire pour tout $\eta_c^t(\delta)$ où δ est compris entre 0 et Δ_c , ce qui définit donc totalement l'état $\eta_c^{t+\Delta_c} : [0, \Delta_c] \rightarrow \mathbb{R}$.



(a) État d'une « portion » de neurone au temps t (b) État d'une « portion » de neurone au temps $t + \Delta_c$

FIGURE 4.17 – Exemple d'un shift de durée Δ_c sur un neurone où tous les compartiments ont le même temps de traversée Δ_c .

L'hypothèse d'égalité de tous les Δ_c que nous avons faite pour des raisons pédagogiques est surtout utile pour que la durée du shift ne dépasse pas la longueur d'un compartiment.

Plus généralement, nous allons maintenant limiter le shift à un petit intervalle de temps de durée ε (Définition 15 ci-après et Figure 4.19). Plus précisément, ε est tel qu'il n'existe aucun compartiment dans le neurone ayant un temps de traversée inférieur à ε . L'amplitude du glissement est limitée à ε pour que l'état de chaque compartiment après le shift puisse être directement calculé à partir de l'état de ses prédécesseurs avant le shift, ce qui évite de lourdes sommes récursives, comme le montre la figure 4.18.

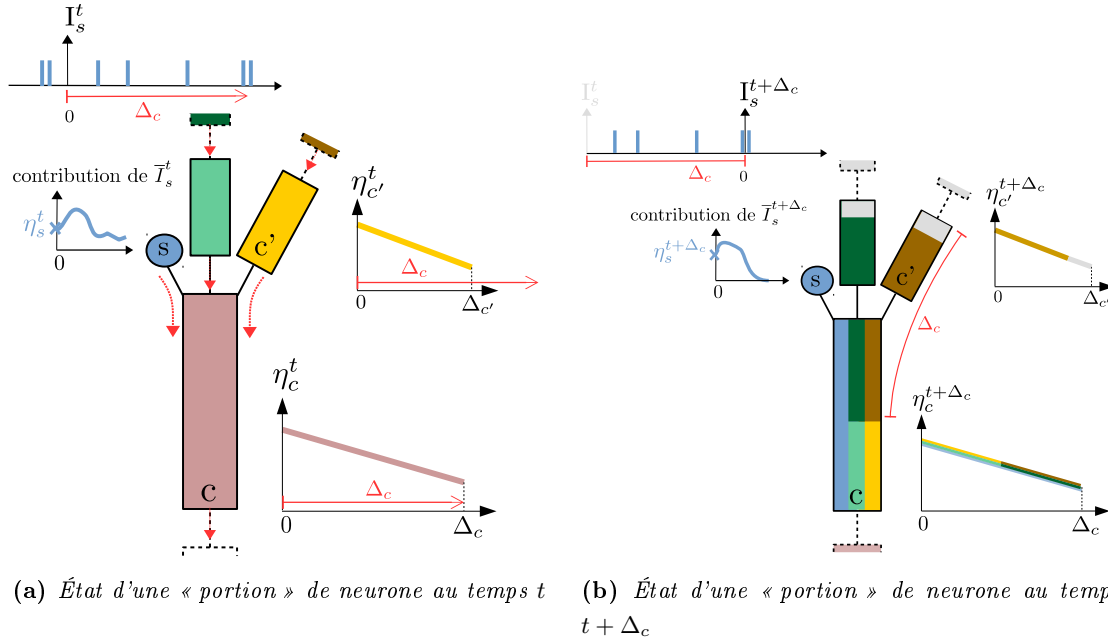


FIGURE 4.18 – Exemple d'un shift d'une durée Δ_c sur un neurone où il existe au moins un compartiment ayant un temps de traversée inférieur à Δ_c (par exemple c'). La valeur de $\eta_c^{t+\Delta_c}(\Delta_c)$ doit être calculée récursivement à partir de l'état des prédécesseurs des prédécesseurs de c au temps t . L'idée sous-jacente est la suivante. La charge en sortie de c au temps $(t + \Delta_c) + \Delta_c$ (soit la valeur de $\eta_c^{t+\Delta_c}(\Delta_c)$) est égale à la charge à l'entrée de c à $t + \Delta_c$ atténué par α_c , c'est à dire à la somme des charges à la sortie des nœuds prédécesseurs de c (y compris c') en même temps atténué par α_c . Or, puisque $\Delta_c > \Delta_{c'}$ par exemple, les charges sortant de c' au temps $t + \Delta_c$ ne se trouvaient pas encore dans c' au temps t , mais dans son compartiment prédécesseur. Ainsi, si on ne limitait pas l'amplitude du shift, il faudrait également considérer dans le calcul, l'état des prédécesseurs des prédécesseurs de c au temps t et ainsi de suite.

Dans la section précédente, nous avons défini l'état d'un neurone. C'est un état tel que l'état d'une synapse est défini, à un temps donné, par un unique nombre réel qui représente la charge à la synapse que l'on suppose ponctuelle. Ici, pour simplifier les définitions futures, nous avons choisi de définir également un *état étendu* dans lequel l'état d'une synapse est un *segment* et non un simple nombre réel (Définition 14 ci-après). Ceci permet d'homogénéiser les notations (comme dans la Définition 15 ci-après, par exemple) puisque l'état d'un compartiment est aussi un segment. Intuitivement, l'état étendu représente non seulement la charge à la synapse à un temps donné, mais aussi la charge dans un intervalle de temps « futur ». De manière similaire à la Définition 13, la Définition 14 ci-après ne vérifie pas la cohérence entre le « passé » du signal d'entrée et l'état du neurone, car la condition de

continuité en zéro suffit pour prouver nos théorèmes.

Définition 14. [Extension d'un état]

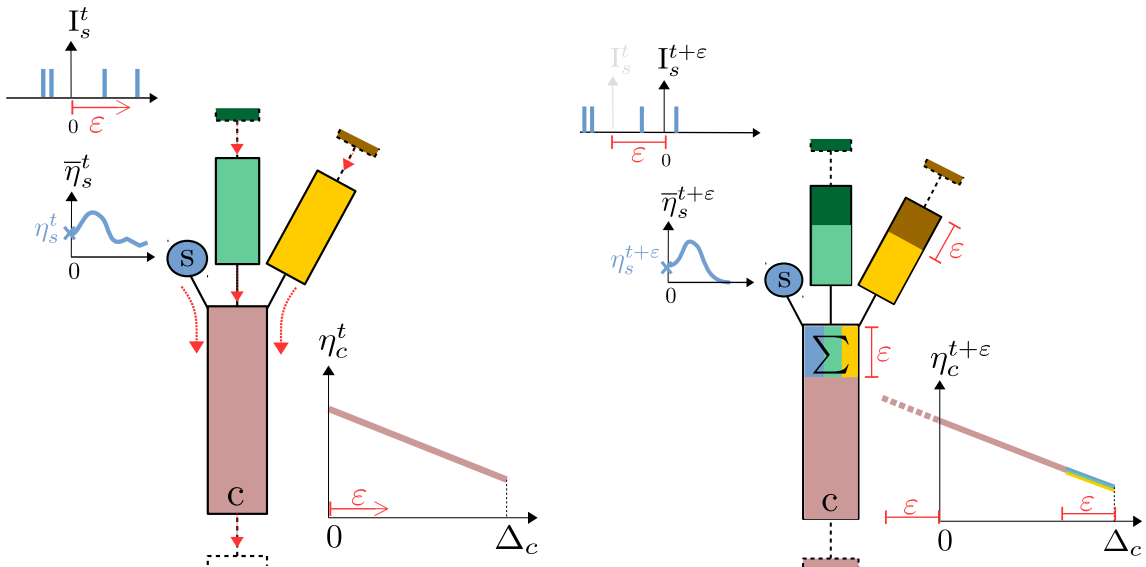
Soient un neurone N , un signal d'entrée $I = \{I_s\}_{s \in Sy(N)}$ et un état $\eta = (\{\eta_s\}_{s \in Sy(N)}, \{\eta_c\}_{c \in Co(N)}, \eta_\nabla)$ de N en continuité avec I . Étant donné $\varepsilon \in \mathbb{R}^{*+}$, on définit l' ε -extension de η conformément à I , $\bar{\eta} = (\{\bar{\eta}_s\}_{s \in Sy(N)}, \{\bar{\eta}_c\}_{c \in Co(N)}, \bar{\eta}_\nabla)$, telle que :

- $\forall s \in Sy(N), \bar{\eta}_s = \bar{I}_s|_{[0, \varepsilon]}$
- $\forall c \in Co(N), \bar{\eta}_c = \eta_c$
- $\bar{\eta}_\nabla = \eta_\nabla$

Soit $\partial = (\{\eta_s\}_{s \in Sy(N)}, \{\eta_c\}_{c \in Co(N)})$, la restriction dendritique de η , on définit de manière similaire, $\bar{\partial}$, l' ε -extension de ∂ conformément à I .

Remarque: Les extensions de η_N conformément à I préservent la continuité avec I puisque $\bar{\eta}_s(0) = \eta_s$. Les extensions simplifient les notations, entre autres, elles simplifient commodément la Définition 8 : $\eta_c(\Delta_c) = \left(\sum_{x \in Pred_N(c)} \bar{\eta}_x(0) \right) \times \alpha_c$.

La définition suivante, en tirant profit de la définition précédente, formalise le shift sur un état dendritique d'un neurone. On remarque que ce shift est bien limité à une durée inférieure ou égale au plus petit temps de traversée des compartiments. On appelle dès lors ε -shift, un shift respectant cette condition (Figure 4.19).



(a) État d'une « portion » de neurone au temps t (b) État d'une « portion » de neurone au temps $t + \varepsilon$

FIGURE 4.19 – Dynamique des dendrites par ε -shift. L'état ($\eta_c^{t+\varepsilon}$) d'un compartiment c au temps $t + \varepsilon$ est calculé à partir de son état (η_c^t) au temps t par glissement. La charge à l'entrée du compartiment c est la somme des charges en sortie de ses nœuds prédécesseurs.

Définition 15. [Shift d'un état dendritique]

Soient un neurone N , un signal d'entrée $I = \{I_s\}_{s \in Sy(N)}$ et soit $\varepsilon \in \mathbb{R}^{*+}$ tel que $\varepsilon \leq \inf(\{\Delta_c \mid c \in Co(N)\})$. On définit $\varepsilon\text{-shift}^I : \mathcal{D}_N \rightarrow \mathcal{D}_N$ comme l'application qui associe à tout $\partial = (\{\eta_s\}_{s \in Sy(N)}, \{\eta_c\}_{c \in Co(N)})$, $\varepsilon\text{-shift}^I(\partial) = (\{\eta'_s\}_{s \in Sy(N)}, \{\eta'_c\}_{c \in Co(N)})$ défini comme suit. Soit $\bar{\partial} = (\{\bar{\eta}_s\}_{s \in Sy(N)}, \{\bar{\eta}_c\}_{c \in Co(N)})$ l' ε -extension de ∂ conformément à I ,

- pour toute synapse $s \in Sy(N)$, $\eta'_s = \bar{\eta}_s(\varepsilon)$;
- pour tout compartiment $c \in Co(N)$, le segment $\eta'_c : [0, \Delta_c] \rightarrow \mathbb{R}$ est défini par :
 - $\forall \delta \in [0, \Delta_c - \varepsilon]$, $\eta'_c(\delta) = \eta_c(\delta + \varepsilon)$;
 - $\forall \delta \in [\Delta_c - \varepsilon, \Delta_c]$, $\eta'_c(\delta) = \left(\sum_{x \in Pred_N(c)} \bar{\eta}_x(\delta - \Delta_c + \varepsilon) \right) \times \alpha_c$.

Remarque: D'après les Définitions 8 et 14, ∂'_N est effectivement un état dendritique, car pour tout $c \in Co(N)$, puisque $\Delta_c \in [\Delta_c - \varepsilon, \Delta_c]$, on a $\forall t \in \mathbb{R}^+$:

$$\eta_c^{t+\varepsilon}(\Delta_c) = \left(\sum_{x \in Pred_N(c)} \bar{\eta}_x^t(\varepsilon) \right) \times \alpha_c = \left(\sum_{x \in Pred_N(c)} \bar{\eta}_x^{t+\varepsilon}(0) \right) \times \alpha_c$$

Par conséquent, si pour tout $x \in Pred_N(c)$ $\bar{\eta}_x^t$ est continu pour tout $c \in Co(N)$ alors, $\eta_c^{t+\varepsilon}$ est aussi continu (condition de continuité, Définition 8).

Notation 2. La famille composée des ε -shifts de tous les η_s conformément à I , est notée $\varepsilon\text{-shift}^I(\{\eta_s\})$. De même, la famille composée des ε -shifts de tous les η_c conformément à I , est notée $\varepsilon\text{-shift}^I(\{\eta_c\})$.

Grâce à la notion d' ε -shift, il est possible de calculer l'état dendritique d'un neurone à partir de son état à un petit intervalle de temps plus tôt. Le Théorème 1 ci-après montre qu'à partir d'un état dendritique donné, une succession d' ε -shifts permet de calculer de manière unique l'état dendritique, après n'importe quel délai. On se rend bien compte de la portée de ce théorème : en appliquant successivement des ε -shifts, on peut donc calculer récursivement, et de manière unique la dynamique des dendrites.

Théorème 1. [Dynamique dendritique]

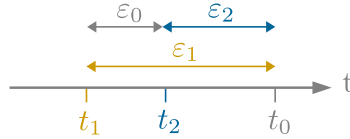
Étant donné un neurone N , soit $\partial^0 = (\{\eta_s^0\}_{s \in Sy(N)}, \{\eta_c^0\}_{c \in Co(N)})$ un état dendritique initial de N en continuité avec un signal d'entrée $I = \{I_s\}_{s \in Sy(N)}$. Il existe une unique famille $\{\partial^t\}_{t \in \mathbb{R}^+}$ telle que pour tout t et pour tout $\varepsilon \in]0, \inf(\{\Delta_c \mid c \in Co(N)\})$:

$$\partial^{t+\varepsilon} = \varepsilon\text{-shift}^I(\partial^t)$$

4.2. DESCRIPTION DE LA DYNAMIQUE D'UN NEURONE

Preuve: Commençons par prouver l'unicité de la famille $\{\partial^t\}$. Supposons que $\{\partial^t\}$ et $\{\partial^{t'}\}$ sont deux familles satisfaisant le Théorème 1. Pour tout $t \in \mathbb{R}^+$, prouvons $\psi(t) : \partial^t = \partial^{t'}$ grâce au Lemme 2 (Chapitre 2, Section 2.1), avec $a = \inf(\{\Delta_c \mid c \in Co(N)\})$. Pour tout $t_0 \in \mathbb{R}^+$, supposons que pour tout $t \in [0, t_0[$, $\partial^t = \partial^{t'}$. D'après le Théorème 1 qui définit $\partial^{t+\varepsilon}$ de manière unique pour tout $\varepsilon \in [0, a]$, on en déduit que pour tout $t \in [t_0, t_0 + a[$, $\partial^t = \partial^{t'}$. Donc pour tout $t \in \mathbb{R}^+$, $\partial^t = \partial^{t'}$.

Il reste à prouver l'existence de la famille $\{\partial^t\}_{t \in \mathbb{R}^+} = (\{\eta_s^t\}_{s \in Sy(N)}, \{\eta_c^t\}_{c \in Co(N)})$. C'est équivalent à prouver que pour tout t_0 , la définition de ∂^{t_0} ne dépend pas de sa décomposition. Donc, supposons $t_0 = t_1 + \varepsilon_1 = t_2 + \varepsilon_2$, il faut prouver : $\varepsilon_2\text{-shift}^I(\partial^{t_2}) = \varepsilon_1\text{-shift}^I(\partial^{t_1})$. On peut toujours supposer que $t_1 < t_2$ et donc $t_2 = t_1 + \varepsilon_0$ où $\varepsilon_0 > 0$. On note $\{\bar{\partial}^t\}_{t \in \mathbb{R}^+} = (\{\bar{\eta}_s^t\}_{s \in Sy(N)}, \{\bar{\eta}_c^t\}_{c \in Co(N)})$, l' ε_1 -extension de $\{\partial^t\}$.



1. ∂^{t_0} en tant que ε_1 -shift de ∂^{t_1} :

(a) Pour toute synapse $s \in Sy(N)$, $\eta_s^{t_0} = \eta_s^{t_1 + \varepsilon_1} = \bar{\eta}_s^{t_1}(\varepsilon_1)$.

(b) Pour tout compartiment $c \in Co(N)$,

i. $\forall \delta \in [0, \Delta_c - \varepsilon_1]$, $\eta_c^{t_0}(\delta) = \eta_c^{t_1 + \varepsilon_1}(\delta) = \eta_c^{t_1}(\delta + \varepsilon_1)$;

ii. $\forall \delta \in [\Delta_c - \varepsilon_1, \Delta_c]$, $\eta_c^{t_0}(\delta) = \eta_c^{t_1 + \varepsilon_1}(\delta) = \left(\sum_{x \in Pred_N(c)} \bar{\eta}_x^{t_1}(\delta - \Delta_c + \varepsilon_1) \right) \times \alpha_c$.

2. ∂^{t_0} en tant que ε_2 -shift de ∂^{t_2} :

(a) Pour toute synapse $s \in Sy(N)$,

$$\eta_s^{t_0} = \eta_s^{t_2 + \varepsilon_2} = \bar{\eta}_s^{t_2}(\varepsilon_2) = \bar{\eta}_s^{t_1 + \varepsilon_0}(\varepsilon_2) = \bar{\eta}_s^{t_1}(\varepsilon_2 + \varepsilon_0) = \bar{\eta}_s^{t_1}(\varepsilon_1)$$

ce qui est égal à $\eta_s^{t_1 + \varepsilon_1}$ (calculé en 1.(a)). Ceci prouve que la définition de $\eta_s^{t_0}$ ne dépend pas de sa décomposition.

(b) Pour tout compartiment $c \in Co(N)$, trois cas (dont l'union couvre l'intervalle $[0, \Delta_c]$) sont à considérer.

i. Si $\delta \in [0, \Delta_c - \varepsilon_1]$ alors a fortiori, $\delta \in [0, \Delta_c - \varepsilon_2]$ car $\varepsilon_2 < \varepsilon_1$. Donc $\eta_c^{t_0}(\delta) = \eta_c^{t_2 + \varepsilon_2}(\delta) = \eta_c^{t_2}(\delta + \varepsilon_2) = \eta_c^{t_1 + \varepsilon_0}(\delta + \varepsilon_2)$. Or, on a, $\delta + \varepsilon_2 \leq \Delta_c - \varepsilon_0$ car $\delta \leq \Delta_c - \varepsilon_1$ avec $\Delta_c - \varepsilon_1 = \Delta_c - (\varepsilon_0 + \varepsilon_2)$. Par conséquent, on a :

$$\begin{aligned} \eta_c^{t_1 + \varepsilon_0}(\delta + \varepsilon_2) &= \eta_c^{t_1}(\delta - \varepsilon_2 + \varepsilon_0) \\ &= \eta_c^{t_1}(\delta + \varepsilon_1) \end{aligned}$$

ce qui est égal à $\eta_c^{t_1 + \varepsilon_1}(\delta)$, calculé en 1(b)i.

ii. Si $\delta \in [\Delta_c - \varepsilon_1, \Delta_c - \varepsilon_2]$ alors a fortiori, $\delta \in [0, \Delta_c - \varepsilon_2]$ car $\varepsilon_1 < \Delta_c$. Donc, $\eta_c^{t_0}(\delta) = \eta_c^{t_2 + \varepsilon_2}(\delta) = \eta_c^{t_2}(\delta + \varepsilon_2) = \eta_c^{t_1 + \varepsilon_0}(\delta + \varepsilon_2)$. Or, on a $\delta + \varepsilon_2 \in [\Delta_c - \varepsilon_0, \Delta_c]$ car $\Delta_c - \varepsilon_1 + \varepsilon_2 = \Delta_c - (\varepsilon_1 - \varepsilon_2) = \Delta_c - \varepsilon_0$.

Donc,

$$\begin{aligned}\eta_c^{t_1+\varepsilon_0}(\delta + \varepsilon_2) &= \left(\sum_{x \in \text{Pred}_N(c)} \bar{\eta}_x^{t_1}(\delta + \varepsilon_2 - \Delta_c + \varepsilon_0) \right) \times \alpha_c \\ &= \left(\sum_{x \in \text{Pred}_N(c)} \bar{\eta}_x^{t_1}(\delta - \Delta_c + \varepsilon_1) \right) \times \alpha_c\end{aligned}$$

ce qui est égal à $\eta_c^{t_1+\varepsilon_1}(t)$, calculé en 1(b)ii. car $\delta \in [\Delta_c - \varepsilon_1, \Delta_c]$.

iii. Si $\delta \in [\Delta_c - \varepsilon_2, \Delta_c]$ alors $\eta_c^{t_2+\varepsilon_2}(\delta) = \left(\sum_{x \in \text{Pred}_N(c)} \bar{\eta}_x^{t_2}(\delta - \Delta_c + \varepsilon_2) \right) \times \alpha_c$.
Or, puisque $\delta \in [\Delta_c - \varepsilon_2, \Delta_c]$ on a, $(\delta - \Delta_c + \varepsilon_2) \in [0, \varepsilon_2] \subset [0, \Delta_c - \varepsilon_0]$ car $\varepsilon_1 = \varepsilon_2 + \varepsilon_0 \leq \Delta_c$ et donc $\varepsilon_2 < \Delta_c - \varepsilon_0$. Donc,

$$\begin{aligned}\bar{\eta}_x^{t_2}(\delta - \Delta_c + \varepsilon_2) &= \bar{\eta}_x^{t_1+\varepsilon_0}(\delta - \Delta_c + \varepsilon_2) \\ &= \bar{\eta}_x^{t_1}(\delta - \Delta_c + \varepsilon_2 + \varepsilon_0) \\ &= \bar{\eta}_x^{t_1}(\delta - \Delta_c + \varepsilon_1)\end{aligned}$$

Il vient que :

$$\eta_c^{t_2+\varepsilon_2}(\delta) = \left(\sum_{x \in \text{Pred}_N(c)} \bar{\eta}_x^{t_1}(\delta - \Delta_c + \varepsilon_1) \right) \times \alpha_c$$

ce qui est égal à $\eta_c^{t_1+\varepsilon_1}(\delta)$ calculé en 1(b)ii, car $\delta \in [\Delta_c - \varepsilon_1, \Delta_c]$.

Donc, $\forall \delta \in [0, \Delta_c]$, $\eta_c^{t_0}$ en tant que ε_2 -shift de $\eta_c^{t_2}$ (calculé en 2.) est égal à $\eta_c^{t_0}$ en tant que ε_1 -shift de $\eta_c^{t_1}$ (calculé en 1.) ce qui prouve que la définition de $\eta_c^{t_0}$ ne dépend pas de sa décomposition.

□

4.2.4 Intégration des signaux : la dynamique du soma

Après avoir parcouru la forêt dendritique, les charges atteignent le soma. La fonction du soma est essentiellement d'intégrer toutes ces charges pour produire une sortie unique. Comme mentionné dans la section 4.1 consacrée à la description statique de notre cadre de modélisation, nous choisissons de modéliser le soma similairement à un neurone de type Leaky Integrate-and-Fire (Chapitre 3, Section 3.1) bien que l'on s'en distingue notamment par la gestion des périodes réfractaires et la mise à jour de la valeur du potentiel lors de l'émission des spikes.

Pour rappel, le modèle Leaky Integrate-and-Fire est régi par l'équation suivante : $\frac{dU}{dt} = \frac{I(t)}{C} - \frac{1}{RC} \cdot U(t)$. Informellement, la variation du voltage U sur un petit intervalle de temps dt dépend de l'intensité du courant entrant dans le neurone $I(t)$, de la capacité de la membrane

4.2. DESCRIPTION DE LA DYNAMIQUE D'UN NEURONE

C ainsi que d'un facteur de fuite $\frac{1}{RC}$ qui s'applique à la valeur courante du voltage (où R est la résistance de la membrane). Nous modélisons le soma grâce à une équation similaire :

$$\frac{dp}{dt} = F(t) - \gamma \cdot p(t) \quad (4.1)$$

La variation de la charge au soma p dépend des charges entrant dans le soma $F(t)$ et d'un facteur de fuite γ appliquée à la valeur courante de la charge au soma, la faisant retourner vers sa valeur de repos.

Lorsque l'on raisonne en dimension, on constate que ces deux équations sont compatibles. Comme énoncé en introduction de ce chapitre, nous savons que la charge est proportionnelle au voltage : $\frac{dp}{dt}$ est donc proportionnel à $\frac{dU}{dt}$ de même que $p(t)$ est proportionnel à $U(t)$. De plus, la charge au soma (appelée potentiel par abus de langage) est ici une grandeur abstraite qui peut donc être directement assimilée au voltage au soma. Notre facteur de fuite γ est un paramètre abstrait qui représente l'inverse de la constante de temps $\tau = RC$ dans le modèle Leaky Integrate-and-Fire. Enfin, notre fonction $F(t)$ est bien compatible avec l'expression $\frac{I(t)}{C}$. Effectivement, l'intensité $I(t)$ décrit un débit de charges entrant dans le neurone et on a $I(t) = \frac{dQ}{dt}$ où dQ est la charge électrique entrante. Ceci signifie que $I(t)$ est proportionnelle aux charges entrantes et donc à la grandeur que nous notons $F(t)$.

L'équation 4.1 nous permet de décrire l'évolution du potentiel de soma au cours du temps en régime sous-liminaire, c'est-à-dire sur des périodes de temps où le potentiel reste sous la valeur du seuil. Au soma, en plus de considérer la valeur du potentiel p , nous tenons également compte du temps écoulé depuis l'émission du dernier spike e afin de gérer les périodes réfractaires. La dynamique globale du soma est la suivante. À partir d'une condition initiale (e_0, p_0) , le potentiel évolue suivant sa dérivée (Équation 4.1). Lorsque p atteint la fonction de borne β caractéristique du soma, c'est-à-dire son seuil, la valeur de e est mise à 0 et la valeur de p est mise à jour, par soustraction de la valeur du seuil p_β . Immédiatement, p suit de nouveau sa dérivée partant d'un nouveau couple (e, p) pour condition initiale. Le Lemme 3 (illustré par la Figure 4.20) décrit l'évolution globale du couple (e, p) au cours du temps, en tenant compte des discontinuités provoquées par l'atteinte du seuil.

Lemme 3. [Lemme technique]

Étant donné un soma $\nabla = (\beta, \gamma)$, il existe une unique famille de fonctions $P_F : \text{Nominal}(\beta) \times \mathbb{R}^+ \rightarrow \mathbb{R}$ indexée par l'ensemble des fonctions continues $F : \mathbb{R}^+ \rightarrow \mathbb{R}$, telle que pour tout couple $(e^0, p^0) \in \text{Nominal}(\beta)$, P_F satisfait :

- $P_F(e^0, p^0, 0) = p^0$;
- $\forall t \in \mathbb{R}^+$, la dérivée à droite $\frac{dP_F(e^0, p^0, t)}{dt}$ existe et est égale à $F(t) - \gamma \cdot P_F(e^0, p^0, t)$;
- $\forall t \in \mathbb{R}^{+*}$, $\ell = \lim_{h \rightarrow t^-} (P_F(e^0, p^0, h))$ existe et :
 - si $(t + e^0, \ell) \in \text{Nominal}(\beta)$, $P_F(e^0, p^0, t)$ est dérivable et $P_F(e^0, p^0, t) = \ell$;
 - sinon, $\forall h \geq t$, $P_F(e^0, p^0, h) = P_G(0, \ell - p_\beta, h - t)$ où G est définie par : $\forall u \in \mathbb{R}^+$, $G(u) = F(u + t)$.

CHAPITRE 4. DÉFINITION DU CADRE DE MODÉLISATION

Preuve: Considérons la suite strictement croissante de nombres réels positifs $t_0 \dots t_n \dots$ construite comme suit :

Cas de base :

- D'après le théorème de Cauchy-Lipschitz (Chapitre 2, Section 2.1), il existe une unique fonction $f(t)$ telle que $f'(t) = g(t, f(t)) = F(t) - \gamma \cdot f(t)$ et, $f(0) = p^0$ car :
 - g est uniformément Lipschitz-continue en f , c'est-à-dire que la constante de Lipschitz (k) ne dépend pas de t donc pour tout $t \in \mathbb{R}^+$, il existe $k \in \mathbb{R}^{*+}$ tel que $|g(t, x) - g(t, y)| \leq k|x - y|$ c'est à dire tel que $|(F(t) - \gamma \cdot x) - (F(t) - \gamma \cdot y)| \leq k|x - y|$ c'est-à-dire tel que $\gamma|y - x| \leq k|x - y|$ ce qui est évident pour $k = \gamma$;
 - g est continue en t car F est elle même continue en t .
- Soit t_0 le plus petit $t \in \mathbb{R}^+ \cup \{+\infty\}$ tel que $(e^0 + t, f(t)) \notin \text{Nominal}(\beta)$. On a, $t_0 > 0$ car $(e^0, f(0)) = (e^0, p^0) \in \text{Nominal}(\beta)$. Si $t_0 = +\infty$ alors le lemme est prouvé. Sinon, $P_F(e^0, p^0, t_0) = P_G(0, p_1, 0)$. Par conséquent, il existe une unique fonction $P_F(e^0, p^0, t)$ sur l'intervalle $[0, t_0]$ et puisqu'elle est dérivable, $\lim_{h \rightarrow t^-} (P_F(e^0, p^0, h))$ existe sur $]0, t_0]$.

Étape d'induction :

- Inductivement, on suppose que l'existence d'une unique fonction $P_F(e^0, p^0, t)$ est prouvée sur l'intervalle $[0, t_i]$ avec $t_i \in \mathbb{R}^+$ et $P_F(e^0, p^0, t_i) = P_{G_i}(0, p_i, 0)$ où $G_i(u) = F(u + t_i)$. Soit Δt le plus petit $t \in \mathbb{R}^+ \cup \{+\infty\}$ tel que $(e^0 + \Delta t, f(t)) \notin \text{Nominal}(\beta)$. Si $\Delta t = +\infty$ alors le lemme est prouvé. Sinon, on pose $t_{i+1} = t_i + \Delta t$. Nous avons prouvé qu'il existe une unique fonction $P_F(e^0, p^0, t)$ sur l'intervalle $[0, t_{i+1}]$. Puisque cette fonction est dérivable sur $[t_i, t_{i+1}]$, $\lim_{h \rightarrow t^-} (P_F(e^0, p^0, h))$ existe sur cet intervalle et donc elle existe sur $[0, t_{i+1}]$.
- Enfin, pour tout i , Δt est supérieur à e_β et par conséquent, la suite des t_i diverge vers $+\infty$ ce qui prouve ce lemme, d'après le lemme 1.

□

On se distingue de beaucoup de modèles dans lesquels le potentiel est remis à 0 lors de l'émission d'un spike. Ici, pour mettre à jour le potentiel p , nous choisissons de soustraire la valeur du seuil p_β , ce qui n'est équivalent que lorsque la période réfractaire relative est dépassée. On admet ainsi que l'état du neurone (en période réfractaire relative ou non) n'influe pas sur la variation de potentiel provoquée par l'ouverture des canaux responsables du potentiel d'action. Au niveau du soma, notre originalité est aussi de prendre en compte la période réfractaire relative qui est souvent négligée dans les modèles classiques tels que Leaky Integrate-and-Fire, ce qui est plus plausible biologiquement.

Nous aurions aimé définir un modèle de soma plus simple que celui-ci, et notamment, sans l'équation différentielle qui compromet de belles propriétés algébriques (Chapitre 6). Cependant, grâce à celle-ci, nous préservons notre objectif de plausibilité biologique : l'équation différentielle qui gouverne le potentiel de soma est nécessaire pour prendre en compte l'accumulation des charges au cours du temps.

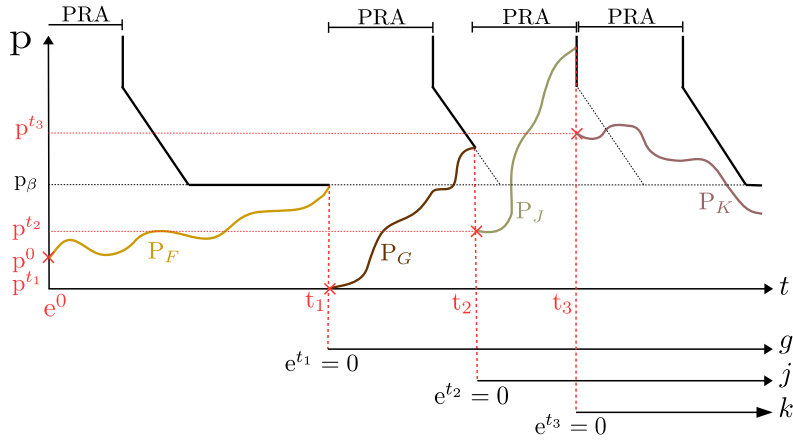


FIGURE 4.20 – Illustration du Lemme 3. Partant d'une condition initiale (e^0, p^0) , la fonction $P_F(e^0, p^0, t)$ décrit l'évolution du couple (e, p) au cours du temps t . Si $t \geq t_1$, $P_F(e^0, p^0, t) = P_G(e^{t_1}, p^{t_1}, g)$ avec $g = t - t_1$; si $t \geq t_2$, $P_F(e^0, p^0, t) = P_J(e^{t_2}, p^{t_2}, j)$ avec $j = t - t_2$; si $t \geq t_3$, $P_F(e^0, p^0, t) = P_K(e^{t_3}, p^{t_3}, k)$ avec $k = t - t_3$. En noir est représentée la fonction de borne β et le sigle PRA désigne la période réfractaire absolue.

4.2.5 Dynamique globale du neurone et signal de sortie discret

La dynamique globale du neurone est l'association de la dynamique de la forêt dendritique avec celle du soma. Lorsqu'un neurone reçoit des spikes au niveau de ses synapses, des mouvements de charges électriques sont générés localement. Ces charges se propagent le long des dendrites compartiment après compartiment, jusqu'à atteindre le soma. Le soma accumule toutes ces charges et si le potentiel résultant est suffisamment important, un spike est émis sur la sortie du neurone.

La définition ci-après décrit comment est calculée la dynamique globale du neurone à partir d'un état initial et d'un signal d'entrée. La dynamique de la forêt dendritique est générée par shifts successifs et la dynamique du soma suit le Lemme 3.

Définition 16. [Dynamique du neurone]

Soit N un neurone et η^0 un état en continuité avec un signal d'entrée $I = \{I_s\}_{s \in Sy(N)}$. La dynamique de N , conformément à I avec η^0 pour état initial, est le segment infini $d_{N,I} : \mathbb{R}^+ \rightarrow \zeta_N$ défini par :

- $d_{N,I}(0) = \eta^0$;
- $\forall t \in \mathbb{R}^+$, $d_{N,I}(t) = \eta^t = (\{\eta_s^t\}_{s \in Sy(N)}, \{\eta_c^t\}_{c \in Co(N)}, (e^t, p^t))$ où :
 - $\{\eta_s^t\}_{s \in Sy(N)}$ et $\{\eta_c^t\}_{c \in Co(N)}$ sont définis tels qu'énoncé par le Théorème 1 ;
 - Considérons préalablement la fonction $F(t) = \sum_{x \in Pred_N(\nabla)} \bar{\eta}_x^t(0)$; F est lipschitzienne, car c'est une combinaison linéaire finie de fonctions lipschitziennes donc d'après le Lemme technique 3, il existe une unique fonction P_F telle que $P_F(e^0, p^0, 0) = p^0$ et pour tout t , $\frac{dP_F(e^0, p^0, t)}{dt} = F(t) - \gamma \cdot P_F(e^0, p^0, t)$.

- * Si $P_F(e^0, p^0, t)$ est continue sur l'intervalle $]0, t]$, alors $e^t = e^0 + t$. Sinon, étant donné t' le plus grand t tel que $P_F(e^0, p^0, t)$ est discontinue, alors $e^t = t - t'$.
- * En considérant la précédente fonction P_F , on a $p^t = P_F(e^0, p^0, t)$.

Au contraire des modèles biophysiques complexes qui décrivent la forme du potentiel d'action, comme c'est le cas du modèle Hodgkin & Huxley (Chapitre 3, Section 3.1), les spikes émis par notre modèle de neurone ne sont pas directement inclus dans la dynamique (Figure 4.21). Ici, les spikes sont seulement déduits de la dynamique et plus précisément de la dynamique du soma.

Définition 17. [Signal de sortie]

Soit N un neurone et $d_I = \{\eta^t\}_{t \in \mathbb{R}^+} = (\{\eta_s^t\}_{s \in S_y(N)}, \{\eta_c^t\}_{c \in C_o(N)}, (e^t, p^t))$ sa dynamique conformément à un signal d'entrée I . Le signal de sortie O de N est le signal élémentaire tel que pour tout $t \in \mathbb{R}^+$:

- si $e^t = 0$ alors $O(t) = 1$,
- sinon $O(t) = 0$.

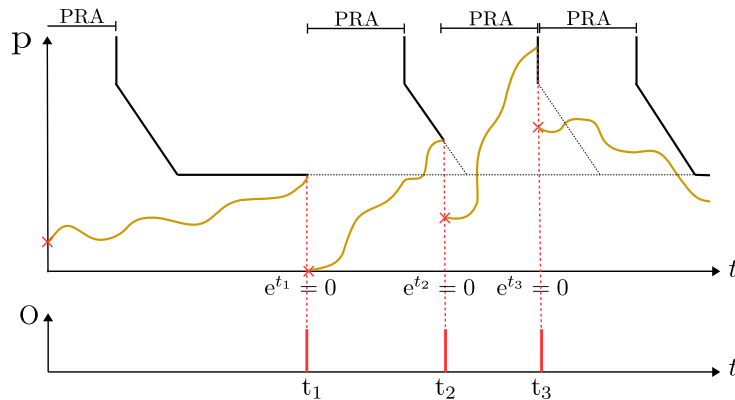


FIGURE 4.21 – Dynamique du soma d'un neurone N . Partant d'une condition initiale (e^0, p^0) , le potentiel de soma p suit sa dérivée. Lorsque p atteint la fonction de borne (ici linéaire par morceaux), un spike est émis sur la sortie (O , en bas de la figure), réinitialisant e à 0, mettant à jour p et déclenchant la période réfractaire absolue (PRA). Ceci est illustré aux temps successifs t_1 , t_2 et t_3 . Puis p continue d'évoluer suivant sa dérivée, partant d'une nouvelle condition initiale.

À partir de la dynamique globale, on a donc défini le signal de sortie du neurone, c'est-à-dire la suite des spikes obtenus en réponse à un signal d'entrée donnée. C'est sur cette fonction d'entrée/sortie qu'on se focalisera dans la suite de la thèse.

CHAPITRE 5

RÉSULTATS FONDAMENTAUX

Dans ce chapitre, nous étudions les propriétés mathématiques du cadre de modélisation que nous avons défini dans le chapitre précédent. On s'intéresse particulièrement aux réductions du modèle de neurone qui préservent sa fonction d'entrée/sortie. L'idée sous-jacente serait de caractériser les classes d'équivalence de structures de neurones partageant le même comportement d'entrée/sortie.

Dans une première section, nous introduisons des définitions et théorèmes relatifs à la réduction. Dans une seconde section, pour aller plus loin, nous discuterons de la normalisation. L'objectif de la normalisation est d'obtenir un neurone minimal permettant de représenter canoniquement une classe d'équivalence, et nous verrons pourquoi ceci reste un problème ouvert. Enfin, nous ferons une interprétation des résultats théoriques énoncés précédemment, du point de vue biologique et du point de vue informatique.

5.1 Les neurones réduits et leurs propriétés

Nous commençons par définir ce que l'on appelle un neurone *réduit*. Un neurone est réduit si et seulement si sa forêt dendritique n'est pas arborescente, c'est-à-dire qu'elle ne contient aucun embranchement (Figure 5.1).

Définition 18. [Neurone réduit]

Un neurone $N \in \mathcal{N}$ est un neurone réduit si et seulement si l'arbre sous-jacent a une hauteur de 1 ou 2. Ceci signifie que chaque synapse de N est soit directement liée au soma, soit liée au soma par un unique compartiment. On note \mathcal{P} l'ensemble des neurones réduits.

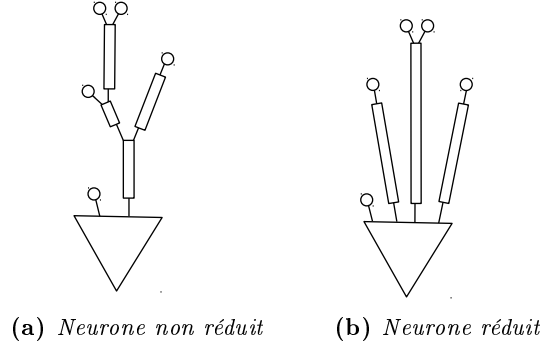


FIGURE 5.1 – Exemples de neurones.

5.1.1 Fonction de réduction

Étant donné un neurone N , nous voulons construire le neurone réduit correspondant, de manière à ce qu'il préserve sa fonction d'entrée/sortie. Pour cela, nous définissons préalablement le *délat* et l'*atténuation* cumulés d'une synapse de N à son soma (Figure 5.2). Plus généralement, les définitions 19 et 20 ci-après permettent de définir le délai et l'atténuation cumulés entre deux nœuds de N .

Définition 19. [Chemin dans un neurone]

Soient un neurone N et deux nœuds distincts $x, y \in \text{Nodes}(N)$. Soit $S = c_1, \dots, c_n$ l'unique suite de compartiments liant x au soma ∇ de N où $c_1 = x$ si $x \in \text{Co}(N)$ et c_1 est le compartiment successeur de x si $x \in \text{Sy}(N)$, et $c_n \in \text{Pred}_N(\nabla)$. Le chemin de x à y dans N , noté $\text{Path}_N(x \rightarrow y)$, est défini comme suit.

- Si $y \in \text{Sy}(N)$ alors $\text{Path}_N(x \rightarrow y)$ n'existe pas.
- Si $y = \nabla$ alors $\text{Path}_N(x \rightarrow y)$ est la suite de compartiments S .
- Si $y \in \text{Co}(N)$:
 - si y est inclus dans S alors $\text{Path}_N(x \rightarrow y)$ est la sous-suite c_1, \dots, y de S ;
 - sinon, $\text{Path}_N(x \rightarrow y)$ n'existe pas.

Définition 20. [Atténuation et délai cumulés]

Étant donné un neurone N et deux nœuds distincts $x, y \in \text{Nodes}(N)$ tels que $\text{Path}_N(x \rightarrow y)$ existe. On définit :

- le délai cumulé de x à y :

$$\Sigma_N^{x \rightarrow y} = \sum_{c_i \in \text{Path}_N(x \rightarrow y)} \Delta_{c_i}$$

- l'atténuation cumulée de x à y :

$$\Pi_N^{x \rightarrow y} = \prod_{c_i \in \text{Path}_N(x \rightarrow y)} \alpha_{c_i}$$

La définition ci-après définit la *fonction de réduction* \mathbb{P} qui associe à tout neurone N un unique neurone réduit $\mathbb{P}(N)$ (Figure 5.2). Le neurone réduit possède les mêmes synapses et le même soma que le neurone d'origine. En revanche, les compartiments peuvent être différents puisque dans le neurone réduit chaque synapse est liée au soma par, au plus, un compartiment. Les paramètres d'un compartiment c liant une synapse s au soma de $\mathbb{P}(N)$ sont respectivement définis par le délai et l'atténuation cumulés de s au soma dans N .

Définition 21. [Fonction de réduction]

La fonction de réduction $\mathbb{P} : \mathcal{N} \rightarrow \mathcal{P}$ est l'application qui associe à tout neurone $N \in \mathcal{N}$, le neurone réduit $\mathbb{P}(N) \in \mathcal{P}$ construit comme suit :

- $\mathbb{P}(N)$ a le même soma que $N : \nabla$;
- $\mathbb{P}(N)$ a le même ensemble de synapses que $N : Sy(\mathbb{P}(N)) = Sy(N)$;
- Pour chaque synapse $s \in Sy(N)$, si $s \in Pred_N(\nabla)$, alors s reste directement liée au soma, sinon l'unique compartiment liant s à ∇ dans $\mathbb{P}(N)$ est le couple $c_s = (\alpha_s, \Delta_s)$ où $\alpha_s = \prod_N^{s \rightarrow \nabla}$ et $\Delta_s = \Sigma_N^{s \rightarrow \nabla}$.

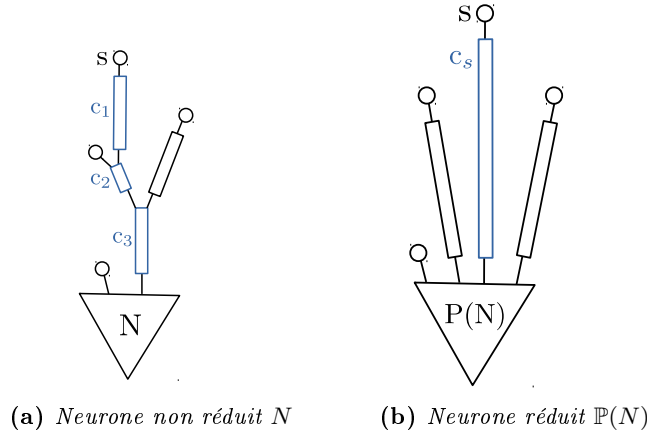


FIGURE 5.2 – *Fonction de réduction.* Le chemin de la synapse s au soma ∇ dans N , noté $Path_N(s \rightarrow \nabla)$, est l'unique suite de compartiments c_1, c_2, c_3 . Le délai cumulé de s à ∇ , noté $\Sigma_N^{s \rightarrow \nabla}$, est égal à la somme $\Delta_{c_1} + \Delta_{c_2} + \Delta_{c_3}$. L'atténuation cumulée de s à ∇ , notée $\prod_N^{s \rightarrow \nabla}$, est égale au produit $\alpha_{c_1} \times \alpha_{c_2} \times \alpha_{c_3}$. La synapse s ainsi que le soma sont les mêmes dans N et dans $\mathbb{P}(N)$. Le compartiment c_s liant s au soma dans $\mathbb{P}(N)$ est défini par le couple $(\prod_N^{s \rightarrow \nabla}, \Sigma_N^{s \rightarrow \nabla})$.

5.1.2 Reconstruction d'état

Nous savons dorénavant construire la forme réduite d'un neurone N *via* la fonction de réduction \mathbb{P} qui permet de calculer les paramètres statiques du neurone réduit $\mathbb{P}(N)$ à partir de ceux du neurone d'origine N .

Ce qui nous intéresse dans ce chapitre est la fonction d'entrée/sortie des neurones que l'on souhaite préserver avec la réduction. Ceci implique de pouvoir comparer la dynamique d'un

neurone avec celle de sa forme réduite, c'est-à-dire d'être capable de comparer leurs états au cours du temps. Dans cette optique, nous proposons dans la Définition 22 ci-après, une fonction permettant de faire le lien entre l'état d'un neurone et celui de sa forme réduite. Cette fonction ($\overleftarrow{\mathbb{P}}_N$) permet de *reconstruire* de manière unique l'état d'un neurone N à partir de l'état de sa forme réduite $\mathbb{P}(N)$. Puisque la réduction conserve les synapses et le soma (Définition 21), la reconstruction de l'état préserve simplement l'état de ces éléments. La complexité de la reconstruction provient de l'état des compartiments (Figure 5.3).

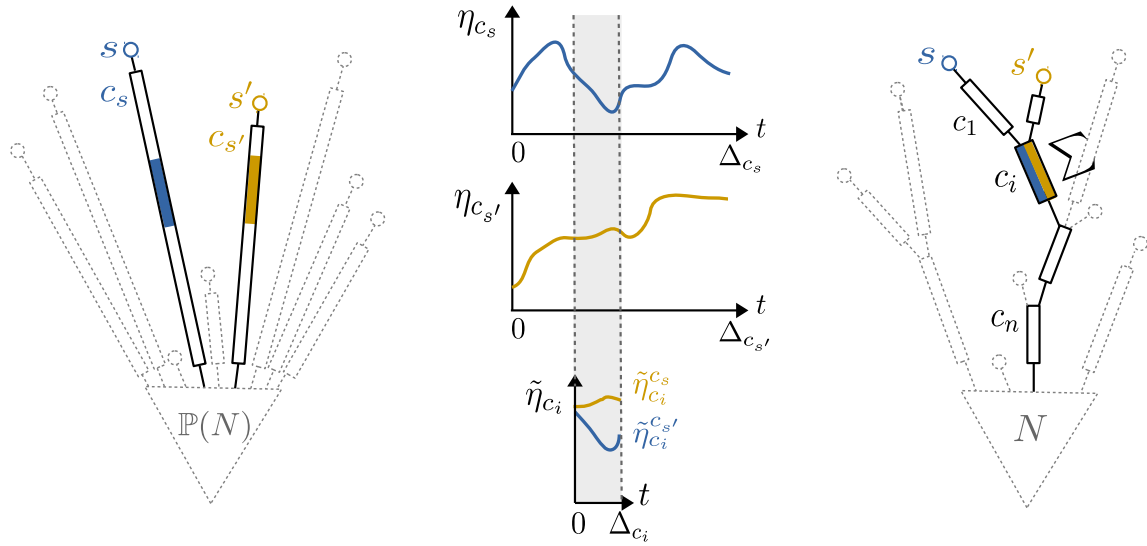


FIGURE 5.3 – *Reconstruction de l'état des compartiments d'un neurone N (à droite) à partir de l'état des compartiments de sa forme réduite $\mathbb{P}(N)$ (à gauche). La partie centrale de cette figure représente l'état $\tilde{\eta}_{c_i}$ du compartiment c_i , calculé à partir des états η_{c_s} et $\eta_{c_{s'}}$ des compartiments c_s et $c_{s'}$: $\tilde{\eta}_{c_i}$ est la somme des c_s - et $c_{s'}$ -contributions à l'état de c_i (respectivement $\tilde{\eta}_{c_i}^{c_s}$ et $\tilde{\eta}_{c_i}^{c_{s'}}$). La contribution des compartiments pontillés de $\mathbb{P}(N)$ à l'état de c_i est nulle.*

Intuitivement, l'état dendritique d'un neurone réduit correspond aux charges présentes en tout point de ses compartiments à un moment donné. La reconstruction de l'état du neurone d'origine consiste finalement à transposer ces charges dans un ensemble de compartiments différents.

La réduction élimine les « troncs communs » des arbres dendritiques pour produire au maximum un compartiment par synapse. La reconstruction, au contraire, tend donc à fusionner les charges des compartiments du neurone réduit dans ces « troncs communs » du neurone d'origine. On remarque que la fonction inverse de la reconstruction ne serait pas déterministe : on ne peut pas construire, de manière unique, l'état d'un neurone réduit à partir de l'état de son neurone d'origine. Il faudrait ajouter un paramètre arbitraire qui représenterait la proportion de charges à transposer dans chaque compartiment provenant de la division d'un « tronc commun » *via* la réduction.

Définition 22. [Reconstruction d'un état]

Étant donné un neurone N , $\overleftarrow{\mathbb{P}}_N : \zeta_{\mathbb{P}(N)} \rightarrow \zeta_N$ est l'application qui associe à tout état $\eta = (\{\eta_s\}_{s \in \text{Sy}(\mathbb{P}(N))}, \{\eta_{c_s}\}_{c_s \in \text{Co}(\mathbb{P}(N))}, \eta_{\nabla})$ de $\mathbb{P}(N)$, l'état $\overleftarrow{\mathbb{P}}_N(\eta) = \tilde{\eta} = (\{\tilde{\eta}_s\}_{s \in \text{Sy}(N)}, \{\tilde{\eta}_c\}_{c \in \text{Co}(N)}, \tilde{\eta}_{\nabla})$ de N défini comme suit :

- $\tilde{\eta}_{\nabla} = \eta_{\nabla}$;
- pour toute synapse $s \in \text{Sy}(N)$, $\tilde{\eta}_s = \eta_s$;
- pour tout compartiment $c_s \in \mathbb{P}(N)$ liant une synapse s au soma de $\mathbb{P}(N)$, on définit préalablement la c_s -contribution de l'état η à l'état $\tilde{\eta}$ comme l'état $\tilde{\eta}^{c_s} = \{\tilde{\eta}_c^{c_s}\}_{c \in \text{Co}(N)}$ tel que pour tout $c \in \text{Co}(N)$, pour tout $\delta \in [0, \Delta_c]$:
 - si $c \notin \text{Path}_N(s \rightarrow \nabla)$ alors :

$$\tilde{\eta}_c^{c_s}(\delta) = 0$$

- sinon, il existe i tel que $c = c_i$ où $\text{Path}_N(s \rightarrow \nabla) = c_1, \dots, c_i, \dots, c_n$ et alors :

$$\tilde{\eta}_c^{c_s}(\delta) = (\eta_{c_s}(\sum_N^{c_{i+1} \rightarrow c_n} + \delta)) / \Pi_N^{c_{i+1} \rightarrow c_n}$$

On définit ensuite, pour tout $c \in \text{Co}(N)$:

$$\tilde{\eta}_c = \sum_{s \in \text{Sy}(\mathbb{P}(N))} \tilde{\eta}_c^{c_s}$$

De manière similaire, on définit $\overleftarrow{\mathbb{P}}_N : \mathcal{D}_{\mathbb{P}(N)} \rightarrow \mathcal{D}_N$, la restriction de $\overleftarrow{\mathbb{P}}$ à la forêt dendritique de N .

Remarque: Si η est en continuité avec un signal d'entrée I , alors $\tilde{\eta}$ est aussi en continuité avec I .

La reconstruction de l'état permet de s'affranchir des considérations techniques relatives à la Définition 13 (Chapitre 4). L'état η de $\mathbb{P}(N)$ est finalement le résultat de l'intégration de son signal d'entrée I . L'état reconstruit $\tilde{\eta}$ de N à partir de l'état η est donc, par construction, cohérente avec le signal d'entrée I passé.

5.1.3 Théorème de réduction

Dans cette sous-section, nous prouvons que la réduction associée à la reconstruction d'état préserve la fonction d'entrée/sortie du neurone d'origine.

Le *théorème de réduction* (Théorème 2 plus loin, illustré par la Figure 5.4) stipule que la dynamique des neurones (Définition 16) commute avec la reconstruction d'état (Définition 22).

En d'autres mots, étant donné un neurone N et sa forme réduite $P = \mathbb{P}(N)$ (ayant des états initiaux « cohérents » et le même signal d'entrée I), on a $d_{N,I}(t) = \overleftarrow{\mathbb{P}}_N(d_{P,I}(t))$ pour tout t . C'est-à-dire que si l'état initial de N est égal à la reconstruction de l'état initial de P alors pour tout temps, l'état de N est égal à la reconstruction de l'état de P . Or, d'après

CHAPITRE 5. RÉSULTATS FONDAMENTAUX

la Définition 22, appliquer $\overleftarrow{\mathbb{P}}_N$ à $d_{P,I}(t)$ (c'est à dire à l'état de P au temps t), ne change pas l'état du soma. Le théorème de réduction implique donc que pour tout temps, l'état du soma de N est égal à l'état du soma de P . De plus, d'après la Définition 21, le soma de N et le soma de P ont les mêmes paramètres statiques. Par conséquent, un neurone N et sa forme réduite P (ayant des états initiaux « cohérents ») ont le même signal de sortie pour tout signal d'entrée commun.

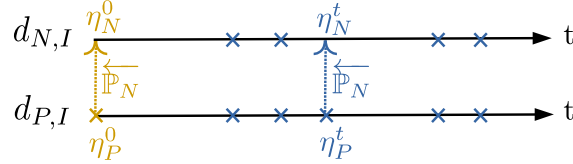


FIGURE 5.4 – Illustration du théorème de réduction. Étant donné un neurone N , sa forme réduite P et un signal d'entrée I , si la dynamique de N au temps 0 est la reconstruction d'état à partir de la dynamique de P au temps 0 (i.e. $d_{N,I}(0) = \eta_N^0 = \overleftarrow{\mathbb{P}}_N(d_{P,I}(0)) = \overleftarrow{\mathbb{P}}_N(\eta_P^0)$) alors, pour tout temps t , la dynamique de N est égale à la reconstruction d'état à partir de la dynamique de P au même temps t (i.e. $d_{N,I}(t) = \eta_N^t = \overleftarrow{\mathbb{P}}_N(d_{P,I}(t)) = \overleftarrow{\mathbb{P}}_N(\eta_P^t)$).

Pour prouver le théorème de réduction, nous allons préalablement prouver deux lemmes :

- le lemme 4 prouve que le shift (Définition 15) commute avec la reconstruction d'état (Définition 22) si le signal d'entrée est restreint à une seule synapse ;
- le lemme 5 prouve que le shift commute également avec la somme (de sorte que l'on pourra sommer les entrées des synapses).

Lemme 4. [Commutativité $shift \leftrightarrow \overleftarrow{\mathbb{P}}_N$]

Soient un neurone N , un état dendritique $\partial = (\{\eta_s\}_{s \in Sy(\mathbb{P}(N))}, \{\eta_{c_s}\}_{c_s \in Co(\mathbb{P}(N))})$ de $\mathbb{P}(N)$ en continuité avec un signal d'entrée I et soit $\varepsilon \in \mathbb{R}^+$ tel que $\varepsilon \leq \inf(\{\Delta_c \mid c \in Co(N)\})$. Pour tout $s \in Sy(\mathbb{P}(N))$:

$$\varepsilon\text{-shift}^{I|_s}(\overleftarrow{\mathbb{P}}_N(\partial)) = \overleftarrow{\mathbb{P}}_N(\varepsilon\text{-shift}^{I|_s}(\partial))$$

Preuve: Nous devons prouver $\varepsilon\text{-shift}^{I|_s}(\overleftarrow{\mathbb{P}}_N(\partial)) = \overleftarrow{\mathbb{P}}_N(\varepsilon\text{-shift}^{I|_s}(\partial))$. Le cas des synapses est immédiat, on se focalise donc sur les compartiments. On commence par noter :

1. $\overleftarrow{\mathbb{P}}_N(\{\eta_{c_s}\}) = \{\overleftarrow{\eta}_c\}_{c \in Co(N)}$
2. $\varepsilon\text{-shift}^{I|_s}(\{\overleftarrow{\eta}_c\}) = \{\overleftarrow{\eta}_c^\varepsilon\}_{c \in Co(N)}$
3. $\varepsilon\text{-shift}^{I|_s}(\{\eta_{c_s}\}) = \{\eta_{c_s}^\varepsilon\}_{c_s \in Co(\mathbb{P}(N))}$
4. $\overleftarrow{\mathbb{P}}_N(\{\eta_{c_s}^\varepsilon\}) = \{\overleftarrow{\eta}_c^\varepsilon\}_{c \in Co(N)}$

Nous devons alors prouver, pour tout $c \in Co(N)$, $\overleftarrow{\eta}_c^\varepsilon = \overleftarrow{\eta}_c^\varepsilon$.

Pour alléger les notations, on note $\eta_s(x) = \overline{\eta}_s(x)$ puisque l'on considère toujours l' ε -extension des états dans cette preuve. De plus, on pose $Path_N(s \rightarrow \nabla) = c_1, \dots, c_i, \dots, c_n$.

1. D'après la Définition 22 :

5.1. LES NEURONES RÉDUITS ET LEURS PROPRIÉTÉS

(a) Si $c \notin \text{Path}_N(s \rightarrow \nabla)$ alors $\forall \delta \in [0, \Delta_c]$, $\overleftarrow{\eta}_c(\delta) = 0$;

(b) Si $c \in \text{Path}_N(s \rightarrow \nabla)$, alors il existe i tel que $c = c_i$.

Donc $\forall \delta \in [0, \Delta_c]$, $\overleftarrow{\eta}_c(\delta) = \eta_{c_s}(\Sigma_N^{c_{i+1} \rightarrow c_n} + \delta) / \Pi_N^{c_{i+1} \rightarrow c_n}$.

2. D'après la Définition 15 :

(a) Si $c \notin \text{Path}_N(s \rightarrow \nabla)$:

i. Si $\delta \in [0, \Delta_c - \varepsilon]$, $\overleftarrow{\eta}_c^\varepsilon(\delta) = \overleftarrow{\eta}_c(\delta + \varepsilon) = 0$ (d'après 1.(a)).

ii. Si $\delta \in [\Delta_c - \varepsilon, \Delta_c]$, $\overleftarrow{\eta}_c^\varepsilon(\delta) = \left(\sum_{x \in \text{Pred}_N(c)} \overleftarrow{\eta}_x(\delta - \Delta_c + \varepsilon) \right) \times \alpha_c$.

Pour chaque $x \in \text{CoPred}_N(c)$, puisque $c \notin \text{Path}_N(s \rightarrow \nabla)$ alors $x \notin \text{Path}_N(s \rightarrow \nabla)$ et donc $\overleftarrow{\eta}_x = 0$ (d'après 1.(a)). De plus, puisque $c \notin \text{Path}_N(s \rightarrow \nabla)$, quelque soit $x \in \text{SyPred}_N(c)$, x est différente de s . Ainsi, pour chaque $x \in \text{SyPred}_N(c)$, puisque l'on considère I_s dans ce lemme, $\overleftarrow{\eta}_x = 0$. On a donc $\overleftarrow{\eta}_c^\varepsilon(\delta) = 0$.

(b) Si $c \in \text{Path}_N(s \rightarrow \nabla)$:

i. Si $\delta \in [0, \Delta_c - \varepsilon]$ (d'après 1.(b)),

$$\begin{aligned} \overleftarrow{\eta}_c^\varepsilon(\delta) &= \overleftarrow{\eta}_c(\delta + \varepsilon) \\ &= \eta_{c_s}(\Sigma_N^{c_{i+1} \rightarrow c_n} + \delta + \varepsilon) / \Pi_N^{c_{i+1} \rightarrow c_n} \end{aligned}$$

ii. Si $\delta \in [\Delta_c - \varepsilon, \Delta_c]$, $\overleftarrow{\eta}_c^\varepsilon(\delta) = \left(\sum_{x \in \text{Pred}_N(c)} \overleftarrow{\eta}_x(\delta - \Delta_c + \varepsilon) \right) \times \alpha_c$.

Si $c_i = c_1$ alors le compartiment c n'a pas de compartiment prédécesseur. Sinon, si $c_i \neq c_1$, alors le compartiment c a un unique compartiment prédécesseur x tel que $x \in \text{Path}_N(s \rightarrow \nabla)$, c'est à dire un unique compartiment prédécesseur ayant potentiellement $\overleftarrow{\eta}_x \neq 0$; plus précisément, $x = c_{i-1}$. De plus, puisque l'on considère I_s dans ce lemme, le compartiment c a au maximum une unique synapse prédécesseuse $x \in \text{SyPred}_N(c)$ ayant potentiellement $\overleftarrow{\eta}_x \neq 0$. Si $c_i \neq c_1$ alors $s \notin \text{SyPred}_N(c)$ et donc pour toute synapse $x \in \text{SyPred}_N(c)$, $\overleftarrow{\eta}_x = 0$. Si $c_i = c_1$ alors le compartiment c a une unique synapse prédécesseuse $x \in \text{SyPred}_N(c)$ ayant potentiellement $\overleftarrow{\eta}_x \neq 0$; plus précisément, $x = s$. Par conséquent :

A. Si $c_i \neq c_1$, $\overleftarrow{\eta}_c^\varepsilon(\delta) = \overleftarrow{\eta}_{c_{i-1}}(\delta - \Delta_c + \varepsilon) \times \alpha_c$.

D'après 1. (b), $\overleftarrow{\eta}_{c_{i-1}}(\delta) = \eta_{c_s}(\Sigma_N^{c_i \rightarrow c_n} + \delta) / \Pi_N^{c_i \rightarrow c_n}$. Donc,

$$\begin{aligned} \overleftarrow{\eta}_c^\varepsilon(\delta) &= (\eta_{c_s}(\Sigma_N^{c_i \rightarrow c_n} + \delta - \Delta_c + \varepsilon) / \Pi_N^{c_i \rightarrow c_n}) \times \alpha_c \\ &= \eta_{c_s}(\Sigma_N^{c_{i+1} \rightarrow c_n} + \delta + \varepsilon) / \Pi_N^{c_{i+1} \rightarrow c_n} \end{aligned}$$

B. Si $c_i = c_1$, $\overleftarrow{\eta}_c^\varepsilon(\delta) = (\overleftarrow{\eta}_s(\delta - \Delta_c + \varepsilon)) \times \alpha_c$ et d'après la Définition 22,

$$\overleftarrow{\eta}_c^\varepsilon(\delta) = (\eta_s(\delta - \Delta_c + \varepsilon)) \times \alpha_c$$

3. D'après la Définition 15 :

(a) Si $\delta \in [0, \Delta_{c_s} - \varepsilon]$, $\eta_{c_s}^\varepsilon(\delta) = \eta_{c_s}(\delta + \varepsilon)$.

(b) Si $\delta \in [\Delta_{c_s} - \varepsilon, \Delta_{c_s}]$, $\eta_{c_s}^\varepsilon(\delta) = \left(\sum_{x \in \text{Pred}_N(c_s)} \eta_x(\delta - \Delta_{c_s} + \varepsilon) \right) \times \alpha_{c_s}$.
 Puisque $c_s \in \text{Co}(\mathbb{P}(N))$, $\text{CoPred}_N(c_s) = \emptyset$ et $\text{SyPred}_N(c_s) = \{s\}$. Donc,

$$\eta_{c_s}^\varepsilon(\delta) = (\eta_s(\delta - \Delta_{c_s} + \varepsilon)) \times \alpha_{c_s}$$

4. D'après la Définition 22 :

(a) Si $c \notin \text{Path}_N(s \rightarrow \nabla)$ alors $\forall \delta \in [0, \Delta_c]$, $\overleftarrow{v}_c^\varepsilon(\delta) = 0$
ce qui est égal à $\overleftarrow{\eta}_c^\varepsilon(\delta)$ (calculé en 2.(a)).

(b) Si $c \in \text{Path}_N(s \rightarrow \nabla)$, il existe i tel que $c = c_i$.
 Donc, $\forall \delta \in [0, \Delta_c]$, $\overleftarrow{\eta}_c^\varepsilon(\delta) = \eta_{c_s}^\varepsilon(\Sigma_N^{c_{i+1} \rightarrow c_n} + \delta) / \Pi_N^{c_{i+1} \rightarrow c_n}$.
 Puisque $\eta_{c_s}^\varepsilon = \varepsilon\text{-shift}^s(\eta_{c_s})$, on a :

i. Si $\delta \in [0, \Delta_c - \varepsilon]$ alors $(\Sigma_N^{c_{i+1} \rightarrow c_n} + \delta) \in [0, \Delta_{c_s} - \varepsilon]$ et donc,
 $\eta_{c_s}^\varepsilon(\Sigma_N^{c_{i+1} \rightarrow c_n} + \delta) = \eta_{c_s}(\Sigma_N^{c_{i+1} \rightarrow c_n} + \delta + \varepsilon)$. Par conséquent,

$$\overleftarrow{\eta}_c^\varepsilon(\delta) = \eta_{c_s}(\Sigma_N^{c_{i+1} \rightarrow c_n} + \delta + \varepsilon) / \Pi_N^{c_{i+1} \rightarrow c_n}$$

ce qui est égal à $\overleftarrow{\eta}_c^\varepsilon(\delta)$ (calculé en 2.(b)i.).

ii. Si $\delta \in [\Delta_c - \varepsilon, \Delta_c]$ alors :

A. Si $c_i \neq c_1$, $(\Sigma_N^{c_{i+1} \rightarrow c_n} + \delta) \in [0, \Delta_{c_s} - \varepsilon]$ et donc,
 $\eta_{c_s}^\varepsilon(\Sigma_N^{c_{i+1} \rightarrow c_n} + \delta) = \eta_{c_s}(\Sigma_N^{c_{i+1} \rightarrow c_n} + \delta + \varepsilon)$. Par conséquent,

$$\overleftarrow{\eta}_c^\varepsilon(\delta) = \eta_{c_s}(\Sigma_N^{c_{i+1} \rightarrow c_n} + \delta + \varepsilon) / \Pi_N^{c_{i+1} \rightarrow c_n}$$

ce qui est égal à $\overleftarrow{\eta}_c^\varepsilon(\delta)$ (calculé en 2.(b)ii.A.).

B. Si $c_i = c_1$, $(\Sigma_N^{c_{i+1} \rightarrow c_n} + \delta) \in [\Delta_{c_s} - \varepsilon, \Delta_{c_s}]$ et donc,

$$\eta_{c_s}^\varepsilon(\Sigma_N^{c_{i+1} \rightarrow c_n} + \delta) = \left(\sum_{x \in \text{Pred}_N(c_s)} \eta_x(\Sigma_N^{c_{i+1} \rightarrow c_n} + \delta - \Delta_{c_s} + \varepsilon) \right) \times \alpha_{c_s}.$$

Puisque $\text{CoPred}_N(c_s) = \emptyset$ et $\text{SyPred}_N(c_s) = \{s\}$, on a

$$\eta_{c_s}^\varepsilon(\Sigma_N^{c_{i+1} \rightarrow c_n} + \delta) = \eta_s(\Sigma_N^{c_{i+1} \rightarrow c_n} + \delta - \Delta_{c_s} + \varepsilon) \times \alpha_{c_s}$$

et par conséquent, $\overleftarrow{\eta}_c^\varepsilon(\delta) = (\eta_s(\Sigma_N^{c_{i+1} \rightarrow c_n} + \delta - \Delta_{c_s} + \varepsilon) \times \alpha_{c_s}) / \Pi_N^{c_{i+1} \rightarrow c_n}$.

Puisque $c_i = c_1$, on a aussi $\Sigma_N^{c_{i+1} \rightarrow c_n} = \Delta_{c_s} - \Delta_{c_1}$ et donc,

$$\overleftarrow{\eta}_c^\varepsilon(\delta) = (\eta_s(\delta - \Delta_{c_1} + \varepsilon) \times \alpha_{c_s}) / \Pi_N^{c_{i+1} \rightarrow c_n}. \text{ Par conséquent,}$$

$$\overleftarrow{\eta}_c^\varepsilon(\delta) = (\eta_s(\delta - \Delta_{c_1} + \varepsilon)) \times \alpha_{c_1}$$

ce qui est égal à $\overleftarrow{\eta}_c^\varepsilon(\delta)$ (calculé en 2.(b)ii.B.).

Ceci conclut la preuve. En 4., nous avons considéré deux cas (a) et (b). Le cas (a) a été directement résolu alors que le cas (b) a été divisé en deux sous-cas i. et ii. Le cas i. a été directement résolu alors que le cas ii. a été divisé en deux sous-sous-cas A. et B. qui ont tous deux été résolus. Donc, grâce à ce découpage, nous avons prouvé le lemme pour toutes les conditions. \square

Lemme 5. [Commutativité $shift \leftrightarrow +$]

Étant donné un neurone N , soient $\partial = (\{\eta_s\}_{s \in Sy(N)}, \{\eta_c\}_{c \in Co(N)})$ et $\partial' = (\{\eta'_s\}_{s \in Sy(N)}, \{\eta'_c\}_{c \in Co(N)})$ deux états dendritiques de N en continuité avec un signal d'entrée I . Soient A et A' deux sous-ensembles disjoints de $Sy(N)$ tels que $A \cup A' = Sy(N)$ et soit $\varepsilon \in \mathbb{R}^+$ tel que $\varepsilon \leq \inf(\{\Delta_c \mid c \in Co(N)\})$. On a :

$$\varepsilon\text{-shift}^I(\partial + \partial') = \varepsilon\text{-shift}^{I_A}(\partial) + \varepsilon\text{-shift}^{I_{A'}}(\partial')$$

Preuve: Nous devons prouver $\varepsilon\text{-shift}^I(\partial + \partial') = \varepsilon\text{-shift}^{I_A}(\partial) + \varepsilon\text{-shift}^{I_{A'}}(\partial')$. Le cas des synapses est immédiat, on se focalise donc sur les compartiments.

Notons $\partial'' = \partial + \partial'$: pour tout $c \in Co(N)$, $\eta''_c = \eta_c + \eta'_c$.

1. On considère $\varepsilon\text{-shift}^I(\{\eta_c\} + \{\eta'_c\})$. Pour tout $c \in Co(N)$, en accord avec la Définition 15 :

- Si $\delta \in [0, \Delta_c - \varepsilon]$,

$$\begin{aligned} \varepsilon\text{-shift}^I(\eta_c + \eta'_c)(\delta) &= \varepsilon\text{-shift}^I(\eta''_c)(\delta) \\ &= \eta''_c(\delta + \varepsilon) \\ &= (\eta_c + \eta'_c)(\delta + \varepsilon) \\ &= \eta_c(\delta + \varepsilon) + \eta'_c(\delta + \varepsilon) \end{aligned}$$

- Si $\delta \in [\Delta_c - \varepsilon, \Delta_c]$,

$$\begin{aligned} \varepsilon\text{-shift}^I(\eta_c + \eta'_c)(\delta) &= \varepsilon\text{-shift}^I(\eta''_c)(\delta) \\ &= \left(\sum_{x \in \text{Pred}_N(c)} (\bar{\eta}''_x(\delta - \Delta_c + \varepsilon)) \right) \times \alpha_c \\ &= \left(\sum_{x \in \text{Pred}_N(c)} ((\bar{\eta}_x + \bar{\eta}'_x)(\delta - \Delta_c + \varepsilon)) \right) \times \alpha_c \\ &= \left(\sum_{x \in \text{Pred}_N(c)} (\bar{\eta}_x(\delta - \Delta_c + \varepsilon) + \bar{\eta}'_x(\delta - \Delta_c + \varepsilon)) \right) \times \alpha_c \end{aligned}$$

2. On considère maintenant $\varepsilon\text{-shift}^{I_A}(\{\eta_c\}) + \varepsilon\text{-shift}^{I_{A'}}(\{\eta'_c\})$. Pour tout $c \in Co(N)$, en accord avec la Définition 15 :

- Si $\delta \in [0, \Delta_c - \varepsilon]$,

$$\varepsilon\text{-shift}^{I_A}(\eta_c) + \varepsilon\text{-shift}^{I_{A'}}(\eta'_c) = \eta_c(\delta + \varepsilon) + \eta'_c(\delta + \varepsilon)$$

- Si $\delta \in [\Delta_c - \varepsilon, \Delta_c]$,

$$\begin{aligned} \varepsilon\text{-shift}^{I_A}(\eta_c) + \varepsilon\text{-shift}^{I_{A'}}(\eta'_c) &= \left(\sum_{c' \in \text{CoPred}_N(c)} \bar{\eta}_{c'}(\delta - \Delta_c + \varepsilon) + \sum_{s \in \text{SyPred}_N(c) \cap s \in A} \bar{\eta}_s(\delta - \Delta_c + \varepsilon) \right) \times \alpha_c \end{aligned}$$

$$\begin{aligned}
& + \left(\sum_{c' \in \text{CoPred}_N(c)} \bar{\eta}'_{c'}(\delta - \Delta_c + \varepsilon) + \sum_{s \in \text{SyPred}_N(c) \cap s \in A'} \bar{\eta}_s(\delta - \Delta_c + \varepsilon) \right) \times \alpha_c \\
& = \left(\sum_{c' \in \text{CoPred}_N(c)} \bar{\eta}_{c'}(\delta - \Delta_c + \varepsilon) + \sum_{s \in \text{SyPred}_N(c) \cap s \in A} \bar{\eta}_s(\delta - \Delta_c + \varepsilon) \right) \\
& + \left(\sum_{c' \in \text{CoPred}_N(c)} \bar{\eta}'_{c'}(\delta - \Delta_c + \varepsilon) + \sum_{s \in \text{SyPred}_N(c) \cap s \in A'} \bar{\eta}_s(\delta - \Delta_c + \varepsilon) \right) \\
& = \left(\sum_{c' \in \text{CoPred}_N(c)} (\bar{\eta}_{c'}(\delta - \Delta_c + \varepsilon) + \bar{\eta}'_{c'}(\delta - \Delta_c + \varepsilon)) \right) \\
& + \left(\sum_{s \in \text{SyPred}_N(c)} \bar{\eta}_s(\delta - \Delta_c + \varepsilon) \right) \\
& = \left(\sum_{x \in \text{Pred}_N(c)} (\bar{\eta}_x(\delta - \Delta_c + \varepsilon) + \bar{\eta}'_x(\delta - \Delta_c + \varepsilon)) \right) \times \alpha_c.
\end{aligned}$$

Donc, $\varepsilon\text{-shift}^{I|_A}(\{\eta_c\}) + \varepsilon\text{-shift}^{I|_{A'}}(\{\eta'_c\}) = \varepsilon\text{-shift}^I(\{\eta_c\} + \{\eta'_c\})$ puisque les équations obtenues en 1. et 2. sont les mêmes. Ceci clôt la preuve. \square

Théorème 2. [Théorème de réduction] Soient un neurone N , le neurone réduit $P = \mathbb{P}(N)$ et η^0 un état de P en continuité avec un signal d'entrée I . Si $d_{P,I}$ est la dynamique de P conformément à I avec η^0 pour état initial et si $d_{N,I}$ est la dynamique de N conformément à I avec $\overleftarrow{\mathbb{P}}_N(\eta^0)$ pour état initial, alors pour tout $t \in \mathbb{R}^+$:

$$d_{N,I}(t) = \overleftarrow{\mathbb{P}}_N(d_{P,I}(t))$$

Preuve: Nous devons prouver $d_{N,I}(t) = \overleftarrow{\mathbb{P}}_N(d_{P,I}(t))$ ce qui revient à prouver que pour tout $t \in \mathbb{R}^+$, $\eta_N^t = \overleftarrow{\mathbb{P}}_N(\eta_P^t)$ où $\eta_P^t = (\partial_P^t, \eta_{\nabla(P)}^t)$ est l'état de P à t et $\eta_N^t = (\partial_N^t, \eta_{\nabla(N)}^t)$ est l'état de N à t .

D'après la Définition 22, c'est évident pour l'état du soma. On se focalise donc sur l'état dendritique. D'après l'énoncé du théorème, on a : $\partial_N^0 = \overleftarrow{\mathbb{P}}_N(\partial_P^0)$. D'après le Théorème 1, il suffit alors de prouver que pour tout $t \in \mathbb{R}^+$ et pour tout $\varepsilon \in]0, \inf(\{\Delta_c | c \in \text{Co}(N)\})$, $\partial_N^{t+\varepsilon} = \overleftarrow{\mathbb{P}}_N(\partial_P^{t+\varepsilon})$. On a :

- $\partial_N^{t+\varepsilon} = \varepsilon\text{-shift}^I(\partial_N^t) = \varepsilon\text{-shift}^I(\overleftarrow{\mathbb{P}}_N(\partial_P^t))$
- $\overleftarrow{\mathbb{P}}_N(\partial_P^{t+\varepsilon}) = \overleftarrow{\mathbb{P}}_N(\varepsilon\text{-shift}^I(\partial_P^t))$

Il reste donc à prouver $\varepsilon\text{-shift}^I(\overleftarrow{\mathbb{P}}_N(\partial_P^t)) = \overleftarrow{\mathbb{P}}_N(\varepsilon\text{-shift}^I(\partial_P^t))$, ce qui résulte directement des Lemmes 5 et 4 puisque tout signal d'entrée I est la somme de ses restrictions aux synapses (Chapitre 4, Section 4.2, Définition 1). \square

Dans cette section, nous avons défini une fonction de réduction (Définition 21) qui permet de construire la forme réduite de n'importe quel neurone, tout en préservant sa fonction d'entrée/sortie (Théorème 2). L'arborescence dendritique est alors réduite (outre les synapses) à un ensemble de couples délai/atténuation portés par les éventuels compartiments uniques

liant les synapses au soma. La représentation des neurones sous forme réduite simplifie donc largement leur structure tout en préservant leur comportement. Tout neurone réduit représente une classe d'équivalence de structures partageant la même fonction d'entrée/sortie.

5.2 Les neurones normalisés : propriétés et limitation

Une continuation naturelle du processus de réduction présenté dans la section précédente, serait d'obtenir des neurones « normalisés » tels que chaque neurone normalisé soit l'*unique* représentant canonique d'une fonction d'entrée/sortie donnée et donc de la classe d'équivalence des entrées/sorties identiques. Deux neurones normalisés différents auraient donc au moins un signal d'entrée permettant de les distinguer du point de vue de leur fonction.

En ce sens, les neurones réduits conformes à la Définition 18 ne sont *pas suffisamment réduits* puisque deux neurones réduits différents peuvent avoir la même fonction d'entrée/sortie.

Contre-exemples :

- Prenons deux neurones réduits N et N' . N possède un soma $\nabla = (\beta, \gamma)$ et une seule synapse $s = \sigma_s$ connectée à ∇ par un unique compartiment $c = (\alpha_c, \Delta_c)$. N' possède le même compartiment, c mais connectant, dans ce cas, une synapse $s' = \sigma_{s'}$ à un soma $\nabla' = (\beta', \gamma')$. Si $\sigma_{s'} = 2 \times \sigma_s$ et $\beta_{s'} = 2 \times \beta_s$ alors il est évident que N et N' , bien qu'ils soient différents, ont la même fonction d'entrée/sortie. Effectivement, le poids de la synapse s' qui est deux fois plus important que celui de la synapse s est compensé par le seuil de N' qui est deux fois plus difficile à atteindre. Par conséquent, pour une même entrée, N et N'' ont toujours la même sortie.
- Prenons un autre neurone réduit N'' . N'' possède le même soma ∇ que, N mais possède une synapse $s'' = \sigma_{s''}$ liée à ∇ par un compartiment $c'' = (\alpha_{c''}, \Delta_{c''})$ où $\sigma_{s''} = 2 \times \sigma_s$ et $\alpha_{c''} = \alpha_c/2$. N et N'' , bien qu'ils soient différents, ont alors la même fonction d'entrée/sortie. En effet, le poids de la synapse de N'' est deux fois plus important que celui de la synapse de, N mais l'atténuation due à la traversée du compartiment c'' est également deux fois plus forte que celle provoquée par le compartiment c , ce qui signifie que le signal entrant au soma est le même dans les deux neurones réduits pour une entrée donnée. Par conséquent, la fonction d'entrée/sortie est la même pour N et N'' .

Les exemples ci-dessus nous amènent à définir un neurone *normalisé* comme un neurone réduit (au sens de la Définition 18) dont les compartiments n'atténuent pas les signaux qui les traversent et dont le seuil d'activation (hors périodes réfractaires) vaut 1. Formellement :

Définition 23. [Neurone normalisé]

Un neurone réduit $N \in \mathcal{P}$ est normalisé si et seulement si :

- pour tout compartiment $c \in Co(N)$, $\alpha_c = 1$;
- la fonction de borne β caractérisant le soma de N est normale (au sens de la Définition 5, Figure 5.5).

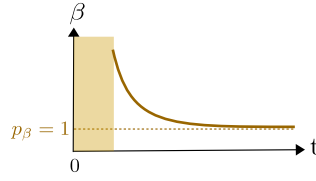


FIGURE 5.5 – Exemple de fonction de borne normale. La fonction de borne β est normale, car $p_\beta = 1$ (Définition 5).

Le seuil d’activation (hors périodes réfractaires) d’un neurone normalisé porte la valeur la plus faible autorisée par la définition de la fonction de borne (Définition 5). Le quotient $\frac{\beta}{\beta'}$ de deux fonctions de borne donne une indication sur leur proximité, comme on le verra plus loin. Ce quotient n’est autre que la fonction constante égale à p_β si β est la fonction de borne d’un neurone et β' la fonction de borne de sa forme normalisée.

5.2.1 Fonction de normalisation

Pour construire la forme normalisée d’un neurone de manière à préserver sa fonction d’entrée/sortie, nous définissons une *fonction de normalisation*. Cette fonction associe à tout neurone N un unique neurone normalisé que l’on note $\mathbb{1}(N)$ (Figure 5.6). Intuitivement, la normalisation du seuil d’activation (hors périodes réfractaires) p_β d’une part, et d’autre part du facteur d’atténuation α des compartiments à 1, est compensée par une diminution du poids des synapses.

Exemple: Prenons un neurone N et sa forme normalisée $\mathbb{1}(N)$. Le neurone N est muni d’un soma $\nabla = (\beta, \gamma)$ et d’une seule synapse $s = \sigma$ connectée à ∇ par un unique compartiment $c = (\alpha, \Delta)$. Le neurone normalisé $\mathbb{1}(N)$ est muni d’un soma $\nabla' = (\beta', \gamma')$ et d’une seule synapse $s' = \sigma'$ connectée à ∇' par un unique compartiment $c' = (\alpha', \Delta')$. Par définition, $p_{\beta'} = 1$. Par conséquent, le seuil de N est p_β fois plus haut que celui de $\mathbb{1}(N)$. Étant donné un même signal d’entrée, pour que le neurone et sa forme normalisée atteignent leur seuil en même temps, le signal arrivant au soma de $\mathbb{1}(N)$ doit donc être p_β fois plus faible que le signal arrivant au soma de N . De plus, puisque par définition $\alpha' = 1$, le signal doit être atténué par α dès l’entrée du compartiment c' , soit dès la synapse s' . La fonction synaptique du neurone normalisé doit donc être adaptée pour prendre en compte la suppression de l’atténuation dans le compartiment et la normalisation du seuil.

Définition 24. [Fonction de normalisation]

La fonction de normalisation $\mathbb{1} : \mathcal{N} \rightarrow \mathcal{P}$ est l’application qui associe à tout neurone $N \in \mathcal{N}$ le neurone normalisé $\mathbb{1}(N) \in \mathcal{P}$ tel que :

- le facteur de fuite γ pour le soma de $\mathbb{1}(N)$ est le même que le facteur de fuite pour le soma de N ;
- la fonction de borne pour le soma de $\mathbb{1}(N)$ est $\beta_{\mathbb{1}} = \frac{\beta}{p_\beta}$ où β est la fonction de borne pour le soma de N ;

5.2. LES NEURONES NORMALISÉS : PROPRIÉTÉS ET LIMITATION

- les ensembles $Sy(\mathbf{1}(N))$ et $Co(\mathbf{1}(N))$ sont définis comme suit :
 - pour chaque synapse $s = \sigma_s$ de $Sy(N)$ directement liée au soma de N , $Sy(\mathbf{1}(N))$ contient la synapse $s = \frac{\sigma_s}{p_\beta}$;
 - pour chaque autre synapse $s = \sigma_s$ de $Sy(N)$, $Sy(\mathbf{1}(N))$ contient la synapse $s = \frac{\sigma_s}{p_\beta} \times \Pi_N^{s \rightarrow \nabla}$ et $Co(\mathbf{1}(N))$ contient le compartiment $c_s = (1, \Sigma_N^{s \rightarrow \nabla})$ liant s au soma de $\mathbf{1}(N)$.

Remarque:

- \mathbb{P} et $\mathbf{1}$ sont idempotentes : $\mathbb{P} \circ \mathbb{P} = \mathbb{P}$ et $\mathbf{1} \circ \mathbf{1} = \mathbf{1}$.
- $\mathbf{1} \circ \mathbb{P} = \mathbf{1}$: pour tout neurone N , $\mathbf{1}(N)$ est égal à $\mathbf{1}(\mathbb{P}(N))$ et donc, il est toujours possible de réduire un neurone avant de le normaliser pour obtenir sa forme normalisée.

Remarque: Étant donné un neurone N , il y a une bijection canonique entre $Sy(N)$, $Sy(\mathbb{P}(N))$ et $Sy(\mathbf{1}(N))$. De plus, étant donné un neurone réduit $P \in \mathcal{P}$, il y a une bijection canonique entre $Co(P)$ et $Co(\mathbf{1}(P))$. En particulier, étant donné un signal d'entrée $I = \{I_s\}_{s \in Sy(N)}$ pour N , I est aussi un signal d'entrée pour $\mathbb{P}(N)$ et pour $\mathbf{1}(N)$.

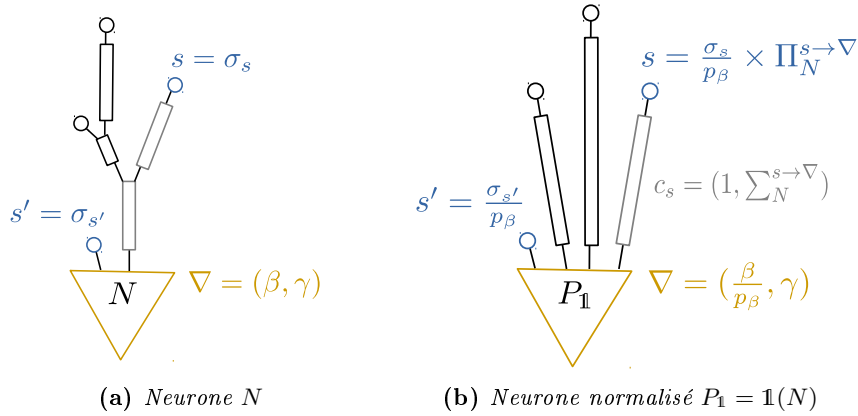


FIGURE 5.6 – Exemple de normalisation. La forme normalisée $P_{\mathbf{1}}$ du neurone N est obtenue par la fonction de normalisation (Définition 24).

5.2.2 Dé-normalisation d'état

La fonction de normalisation permet de définir les paramètres statiques de la forme normalisée d'un neurone donné. Similairement au processus de réduction (Section 5.1), nous allons prouver la préservation de la fonction d'entrée/sortie par la normalisation. Ceci implique de pouvoir comparer l'état des neurones et de leur forme normalisée au cours du temps. Pour ce faire, nous proposons dans la Définition 25 ci-après, une fonction permettant, par composition avec la fonction de reconstruction d'état (Définition 22), de faire le lien entre l'état d'un neurone N et celui de sa forme normalisée $\mathbf{1}(N)$. Cette fonction ($\overleftarrow{\mathbf{1}}_P$) permet

de *dénormaliser* de manière unique l'état de $\mathbf{1}(N)$ pour obtenir l'état de la forme réduite $\mathbb{P}(N)$. Pour obtenir l'état du neurone d'origine N , il suffit de composer cette fonction avec la fonction de reconstruction d'état. Notons que contrairement à la reconstruction d'état, la dé-normalisation possède une fonction inverse qui permet de calculer de manière unique l'état de la forme normalisée à partir de l'état d'un neurone réduit, car ces deux formes ont les mêmes compartiments.

Définition 25. [Dé-normalisation d'un état]

Étant donné un neurone réduit P , on définit $\overleftarrow{\mathbb{I}}_P : \zeta_{\mathbf{1}(P)} \rightarrow \zeta_P$ comme l'application qui associe à tout état $\eta_{\mathbf{1}} = (\{\eta_{s_{\mathbf{1}}}\}_{s_{\mathbf{1}} \in \text{Sy}(\mathbf{1}(P))}, \{\eta_{c_{\mathbf{1}}}\}_{c_{\mathbf{1}} \in \text{Co}(\mathbf{1}(P))}, \eta_{\nabla(\mathbf{1}(P))})$ de $\mathbf{1}(P)$, l'état $\eta_P = (\{\eta_{s_P}\}_{s_P \in \text{Sy}(P)}, \{\eta_{c_P}\}_{c_P \in \text{Co}(P)}, \eta_{\nabla(P)})$ de P tel que :

- pour chaque synapse $s_P \in P$,
 - si s_P est directement liée au soma de P alors $\eta_{s_P} = \eta_{s_{\mathbf{1}}} \times p_{\beta}$;
 - sinon, $\eta_{s_P} = \frac{\eta_{s_{\mathbf{1}}}}{\alpha_{c_P}} \times p_{\beta}$ où $s_{\mathbf{1}} \in \mathbf{1}(P)$ est la synapse correspondant à $s_P \in P$ et c_P est l'unique compartiment liant s_P au soma dans P ;
- pour chaque compartiment $c_P \in P$, $\eta_{c_P} = \eta_{c_{\mathbf{1}}} \times p_{\beta}$ où $c_{\mathbf{1}} \in \mathbf{1}(P)$ est le compartiment correspondant à $c_P \in P$;
- $\eta_{\nabla(P)} = (e_{\mathbf{1}}, p_{\mathbf{1}} \times p_{\beta})$ où $\eta_{\nabla(\mathbf{1}(P))} = (e_{\mathbf{1}}, p_{\mathbf{1}})$.

De plus, on note $\overleftarrow{\mathbb{I}}_N = \overleftarrow{\mathbb{P}}_N \circ \overleftarrow{\mathbb{I}}_{\mathbb{P}(N)}$.

Remarque: Si $\eta_{\mathbf{1}}$ est en continuité avec un signal d'entrée I , alors $\eta_P = \overleftarrow{\mathbb{I}}_P(\eta_{\mathbf{1}})$ est aussi en continuité avec I .

Preuve: Pour chaque synapse $s_P = \sigma_{s_P} \in \text{Sy}(P)$:

- si s_P est directement liée au soma de P alors $\eta_{s_P} = \eta_{s_{\mathbf{1}}} \times p_{\beta}$; or d'après la Définition 24, on a $\sigma_{s_P} = \sigma_{s_{\mathbf{1}}} \times p_{\beta}$ et donc d'après la Définition 12, $\bar{I}_{s_P}(0) = \bar{I}_{s_{\mathbf{1}}}(0) \times p_{\beta}$; par ailleurs si $\eta_{\mathbf{1}}$ est en continuité avec I alors d'après la Définition 13, $\eta_{s_{\mathbf{1}}} = \bar{I}_{s_{\mathbf{1}}}(0)$; par conséquent, $\eta_{s_P} = \bar{I}_{s_{\mathbf{1}}}(0) \times p_{\beta} = \bar{I}_{s_P}(0)$ rendant η_P en continuité avec I ;
- sinon, $\eta_{s_P} = \frac{\eta_{s_{\mathbf{1}}}}{\alpha_{c_P}} \times p_{\beta}$; or d'après la Définition 24, on a $\sigma_{s_P} = \frac{\sigma_{s_{\mathbf{1}}}}{\alpha_{c_P}} \times p_{\beta}$ et donc d'après la Définition 12, $\bar{I}_{s_P}(0) = \frac{\bar{I}_{s_{\mathbf{1}}}(0)}{\alpha_{c_P}} \times p_{\beta}$; par ailleurs, si $\eta_{\mathbf{1}}$ est en continuité avec I alors $\eta_{s_{\mathbf{1}}} = \bar{I}_{s_{\mathbf{1}}}(0)$; par conséquent, $\eta_{s_P} = \frac{\bar{I}_{s_{\mathbf{1}}}(0)}{\alpha_{c_P}} \times p_{\beta} = \bar{I}_{s_P}(0)$ rendant η_P en continuité avec I .

□

Remarque: De plus, d'après la Définition 22, si η_P est en continuité avec I alors $\overleftarrow{\mathbb{P}}_N(\eta_P)$ est aussi en continuité avec I . Par conséquent, si $\eta_{\mathbf{1}}$ est en continuité avec I , $\overleftarrow{\mathbb{I}}_P(\overleftarrow{\mathbb{P}}_N(\eta_P))$ est aussi en continuité avec I .

5.2.3 Théorème de normalisation

Dans cette sous-section, nous prouvons que la normalisation associée à la dé-normalisation d'état préserve la fonction d'entrée/sortie du neurone d'origine.

Nous commençons par prouver, dans le *théorème de normalisation* (Théorème 3 ci-après, illustré par la Figure 5.7) que la dynamique des neurones (Définition 16) commute avec la dé-normalisation d'état (Définition 22). En d'autres mots, étant donné un neurone réduit P et sa forme normalisée $\mathbb{1}(P)$ (ayant des états initiaux « cohérents » et le même signal d'entrée I), on a $d_{P,I}(t) = \overleftarrow{\mathbb{1}}_P(d_{P_{\mathbb{1}},I}(t))$ pour tout t . C'est-à-dire que si l'état initial de P est égal à la dé-normalisation de l'état initial de $\mathbb{1}(P)$ alors pour tout temps, l'état de P est égal à la dé-normalisation de l'état de $\mathbb{1}(P)$.

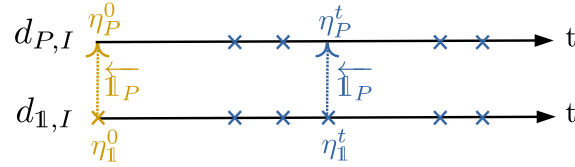


FIGURE 5.7 – Illustration du théorème de normalisation. Étant donné un neurone réduit P , sa forme normalisée $\mathbb{1} = \mathbb{1}(P)$ et un signal d'entrée I , si l'état de P au temps 0 est la dé-normalisation de l'état de $\mathbb{1}$ au temps 0 (i.e. $d_{P,I}(0) = \eta_P^0 = \overleftarrow{\mathbb{1}}_P(d_{\mathbb{1},I}(0)) = \overleftarrow{\mathbb{1}}_P(\eta_{\mathbb{1}}^0)$) alors, pour tout temps t , la dynamique de P est égale à la dé-normalisation d'état à partir de la dynamique de $\mathbb{1}$ au même temps (i.e. $d_{P,I}(t) = \eta_P^t = \overleftarrow{\mathbb{1}}_P(d_{\mathbb{1},I}(t)) = \overleftarrow{\mathbb{1}}_P(\eta_{\mathbb{1}}^t)$).

Théorème 3. [Théorème de normalisation]

Soient un neurone réduit P , le neurone normalisé $P_{\mathbb{1}} = \mathbb{1}(P)$ et η^0 un état de $P_{\mathbb{1}}$ en continuité avec un signal d'entrée I . Si $d_{P_{\mathbb{1}},I}$ est la dynamique de $P_{\mathbb{1}}$ (conformément à I avec η^0 comme état initial) et si $d_{P,I}$ est la dynamique de P (conformément à I avec $\overleftarrow{\mathbb{1}}_P(\eta^0)$ comme état initial), alors pour tout $t \in \mathbb{R}^+$:

$$d_{P,I}(t) = \overleftarrow{\mathbb{1}}_P(d_{P_{\mathbb{1}},I}(t))$$

Preuve: Nous devons prouver $d_{P,I}(t) = \overleftarrow{\mathbb{1}}_P(d_{P_{\mathbb{1}},I}(t))$. Ceci revient à prouver que pour tout t et conformément au signal d'entrée I :

1. $\eta_{s_P}^t = \overleftarrow{\mathbb{1}}_P(\eta_{s_{\mathbb{1}}}^t)$ pour toute synapse s_P de P avec $\overleftarrow{\mathbb{1}}_P(\eta_{s_{\mathbb{1}}}^0)$ comme état initial où $s_{\mathbb{1}}$ est la synapse correspondant à s_P dans $P_{\mathbb{1}}$, ayant $\eta_{s_{\mathbb{1}}}^0$ comme état initial;
2. $\eta_{c_P}^t = \overleftarrow{\mathbb{1}}_P(\eta_{c_{\mathbb{1}}}^t)$ pour tout compartiment c_P de P avec $\overleftarrow{\mathbb{1}}_P(\eta_{c_{\mathbb{1}}}^0)$ comme état initial où $c_{\mathbb{1}}$ est le compartiment correspondant à c_P dans $P_{\mathbb{1}}$, ayant $\eta_{c_{\mathbb{1}}}^0$ comme état initial;
3. $\eta_{\nabla_P}^t = \overleftarrow{\mathbb{1}}_P(\eta_{\nabla_{\mathbb{1}}}^t)$ où ∇_P est le soma de P avec $\overleftarrow{\mathbb{1}}_P(\eta_{\nabla_{\mathbb{1}}}^0)$ comme état initial et $\nabla_{\mathbb{1}}$ est le soma de $P_{\mathbb{1}}$ avec $\eta_{\nabla_{\mathbb{1}}}^0$ comme état initial.

CHAPITRE 5. RÉSULTATS FONDAMENTAUX

Avant tout, étant donné un neurone réduit $P \in \mathcal{P}$, pour tout compartiment $c \in \text{Co}(P)$ liant une synapse s au soma de P , on a $\text{CoPred}_P(c) = \emptyset$ et $\text{SyPred}_P(c) = \{s\}$. On a donc, pour tout $\delta \in [\Delta_c - \varepsilon, \Delta_c]$, $\eta_c^{t+\varepsilon} = (\bar{\eta}_s^t(\delta - \Delta_c + \varepsilon)) \times \alpha_c$ où $\alpha_c = 1$ si P est un neurone normalisé. De plus, pour simplifier les notations, on note $\eta_s(x) = \bar{\eta}_s(x)$ lorsque l'on considère l' ε -extension des états dans cette preuve. On note également β la fonction de borne pour le soma de P et $\beta_{\mathbb{1}}$ la fonction de borne pour le soma de $P_{\mathbb{1}}$.

1. D'après le Théorème 1 et sachant $\eta_{s_P}^0 = \overleftarrow{\mathbb{1}}_P(\eta_{s_{\mathbb{1}}}^0)$, prouver que pour tout t , $\eta_{s_P}^t = \overleftarrow{\mathbb{1}}_P(\eta_{s_{\mathbb{1}}}^t)$ revient à prouver que pour tout $\varepsilon \in \mathbb{R}^{*+}$: $\varepsilon\text{-shift}^I(\overleftarrow{\mathbb{1}}_P(\eta_{s_{\mathbb{1}}})) = \overleftarrow{\mathbb{1}}_P(\varepsilon\text{-shift}^I(\eta_{s_{\mathbb{1}}}))$.
 - (a) On note $\overleftarrow{\eta}_{s_{\mathbb{1}}} = \overleftarrow{\mathbb{1}}_P(\eta_{s_{\mathbb{1}}})$. D'après la Définition 25 :
 - Si $s_{\mathbb{1}}$ est directement liée au soma, $\overleftarrow{\eta}_{s_{\mathbb{1}}} = \eta_{s_{\mathbb{1}}} \times p_{\beta}$.
 - Si $s_{\mathbb{1}}$ est liée au soma par un unique compartiment $c_{\mathbb{1}}$, $\overleftarrow{\eta}_{s_{\mathbb{1}}} = \frac{\eta_{s_{\mathbb{1}}}}{\alpha_{c_{\mathbb{1}}}} \times p_{\beta}$.
 - (b) On note $\overleftarrow{\eta}_{s_{\mathbb{1}}}^{\varepsilon} = \varepsilon\text{-shift}^I(\overleftarrow{\eta}_{s_{\mathbb{1}}})$ et on a $\overleftarrow{\eta}_{s_{\mathbb{1}}}^{\varepsilon} = \overleftarrow{\eta}_{s_{\mathbb{1}}}^0(\varepsilon)$. D'après 1.(a) :
 - Si $s_{\mathbb{1}}$ est directement liée au soma, $\overleftarrow{\eta}_{s_{\mathbb{1}}}^{\varepsilon} = (\eta_{s_{\mathbb{1}}}^0(\varepsilon)) \times p_{\beta}$.
 - Si $s_{\mathbb{1}}$ est liée au soma par un unique compartiment $c_{\mathbb{1}}$, $\overleftarrow{\eta}_{s_{\mathbb{1}}}^{\varepsilon} = \frac{(\eta_{s_{\mathbb{1}}}^0(\varepsilon))}{\alpha_{c_{\mathbb{1}}}} \times p_{\beta}$.
 - (c) On note $\eta_{s_{\mathbb{1}}}^{\varepsilon} = \varepsilon\text{-shift}^I(\eta_{s_{\mathbb{1}}})$ et on a $\eta_{s_{\mathbb{1}}}^{\varepsilon} = \eta_{s_{\mathbb{1}}}^0(\varepsilon)$.
 - (d) On note $\overleftarrow{\eta}_{s_{\mathbb{1}}}^{\varepsilon} = \overleftarrow{\mathbb{1}}_P(\eta_{s_{\mathbb{1}}}^{\varepsilon})$. D'après la Définition 25 et 1.(c) :
 - Si $s_{\mathbb{1}}$ est directement liée au soma, $\overleftarrow{\eta}_{s_{\mathbb{1}}}^{\varepsilon} = (\eta_{s_{\mathbb{1}}}^0(\varepsilon)) \times p_{\beta}$
ce qui est égal à 1.(b).
 - Si $s_{\mathbb{1}}$ est liée au soma par un unique compartiment $c_{\mathbb{1}}$, $\overleftarrow{\eta}_{s_{\mathbb{1}}}^{\varepsilon} = \frac{(\eta_{s_{\mathbb{1}}}^0(\varepsilon))}{\alpha_{c_{\mathbb{1}}}} \times p_{\beta}$
ce qui est égal à 1.(b), prouvant 1.
2. D'après le Théorème 1 et sachant $\eta_{c_P}^0 = \overleftarrow{\mathbb{1}}_P(\eta_{c_{\mathbb{1}}}^0)$, prouver que pour tout t , $\eta_{c_P}^t = \overleftarrow{\mathbb{1}}_P(\eta_{c_{\mathbb{1}}}^t)$ revient à prouver que pour tout $\varepsilon \in \mathbb{R}^{*+}$: $\varepsilon\text{-shift}^I(\overleftarrow{\mathbb{1}}_P(\eta_{c_{\mathbb{1}}})) = \overleftarrow{\mathbb{1}}_P(\varepsilon\text{-shift}^I(\eta_{c_{\mathbb{1}}}))$.
 - (a) On note $\overleftarrow{\eta}_{c_{\mathbb{1}}} = \overleftarrow{\mathbb{1}}_P(\eta_{c_{\mathbb{1}}})$. D'après la Définition 25, $\overleftarrow{\eta}_{c_{\mathbb{1}}} = \eta_{c_{\mathbb{1}}} \times p_{\beta}$.
 - (b) On note $\overleftarrow{\eta}_{c_{\mathbb{1}}}^{\varepsilon} = \varepsilon\text{-shift}^I(\overleftarrow{\eta}_{c_{\mathbb{1}}})$.
 - $\forall \delta \in [0, \Delta_{c_{\mathbb{1}}} - \varepsilon]$, $\overleftarrow{\eta}_{c_{\mathbb{1}}}^{\varepsilon}(\delta) = \overleftarrow{\eta}_{c_{\mathbb{1}}}^0(\delta + \varepsilon)$
ce qui est égal à $(\eta_{c_{\mathbb{1}}}^0(\delta + \varepsilon)) \times p_{\beta}$ d'après 2.(a).
 - $\forall \delta \in [\Delta_{c_{\mathbb{1}}} - \varepsilon, \Delta_{c_{\mathbb{1}}}]$, $\overleftarrow{\eta}_{c_{\mathbb{1}}}^{\varepsilon}(\delta) = \overleftarrow{\eta}_{s_{\mathbb{1}}}^0(\delta - \Delta_{c_{\mathbb{1}}} + \varepsilon)$
ce qui est égal à $(\eta_{s_{\mathbb{1}}}^0(\delta - \Delta_{c_{\mathbb{1}}} + \varepsilon)) \times p_{\beta}$ d'après la Définition 25.
 - (c) On note $\eta_{c_{\mathbb{1}}}^{\varepsilon} = \varepsilon\text{-shift}^I(\eta_{c_{\mathbb{1}}})$.
 - $\forall \delta \in [0, \Delta_{c_{\mathbb{1}}} - \varepsilon]$, $\eta_{c_{\mathbb{1}}}^{\varepsilon}(\delta) = \eta_{c_{\mathbb{1}}}^0(\delta + \varepsilon)$.
 - $\forall \delta \in [\Delta_{c_{\mathbb{1}}} - \varepsilon, \Delta_{c_{\mathbb{1}}}]$, $\eta_{c_{\mathbb{1}}}^{\varepsilon}(\delta) = \eta_{s_{\mathbb{1}}}^0(\delta - \Delta_{c_{\mathbb{1}}} + \varepsilon)$.
 - (d) On note $\overleftarrow{\eta}_{c_{\mathbb{1}}}^{\varepsilon} = \overleftarrow{\mathbb{1}}_P(\eta_{c_{\mathbb{1}}}^{\varepsilon})$. D'après la Définition 25, $\overleftarrow{\eta}_{c_{\mathbb{1}}}^{\varepsilon} = \eta_{c_{\mathbb{1}}}^{\varepsilon} \times p_{\beta}$.
 - $\forall \delta \in [0, \Delta_{c_{\mathbb{1}}} - \varepsilon]$, $\overleftarrow{\eta}_{c_{\mathbb{1}}}^{\varepsilon}(\delta) = (\eta_{c_{\mathbb{1}}}^0(\delta + \varepsilon)) \times p_{\beta}$ d'après 2.(c)
ce qui est égal à 2.(b).
 - $\forall \delta \in [\Delta_{c_{\mathbb{1}}} - \varepsilon, \Delta_{c_{\mathbb{1}}}]$, $\overleftarrow{\eta}_{c_{\mathbb{1}}}^{\varepsilon}(\delta) = (\eta_{s_{\mathbb{1}}}^0(\delta - \Delta_{c_{\mathbb{1}}} + \varepsilon)) \times p_{\beta}$ d'après 2.(c)
ce qui est égal à 2.(b), prouvant 2.
3. D'après la Définition 16 et sachant $\eta_{\nabla_P}^0 = \overleftarrow{\mathbb{1}}_P(\eta_{\nabla_{\mathbb{1}}}^0)$, prouver que pour tout t , $\eta_{\nabla_P}^t = \overleftarrow{\mathbb{1}}_P(\eta_{\nabla_{\mathbb{1}}}^t)$ revient à prouver que pour tout t , $p_P^t = \overleftarrow{\mathbb{1}}_P(p_{\mathbb{1}}^t)$ et $e_P^t = \overleftarrow{\mathbb{1}}_P(e_{\mathbb{1}}^t)$.
D'après la Définition 16, on a :

- $p_P^t = P_{F_P}(\overleftarrow{\mathbb{1}}_P(e_{\mathbb{1}}^0), \overleftarrow{\mathbb{1}}_P(p_{\mathbb{1}}^0), t)$
- $p_{\mathbb{1}}^t = P_{F_{\mathbb{1}}}(e_{\mathbb{1}}^0, p_{\mathbb{1}}^0, t)$.

Étant donné le Lemme 3, il existe :

- Une unique fonction $P_{F_{\mathbb{1}}}$ telle que :
 - $P_{F_{\mathbb{1}}}(e_{\mathbb{1}}^0, p_{\mathbb{1}}^0, 0) = p_{\mathbb{1}}^0$;
 - Pour tout $t \in \mathbb{R}^+$, $\frac{dP_{F_{\mathbb{1}}}(e_{\mathbb{1}}^0, p_{\mathbb{1}}^0, t)}{dt} = F_{\mathbb{1}}(t) - \gamma_{\mathbb{1}} \times P_{F_{\mathbb{1}}}(e_{\mathbb{1}}^0, p_{\mathbb{1}}^0, t)$
où $F_{\mathbb{1}}(t) = \sum_{x_{\mathbb{1}} \in \text{Pred}_{P_{\mathbb{1}}}(\nabla)} \eta_{x_{\mathbb{1}}}^t(0)$;
 - Pour tout $t \in \mathbb{R}^{+*}$, $\ell = \lim_{h \rightarrow t^-} (P_{F_{\mathbb{1}}}(e_{\mathbb{1}}^0, p_{\mathbb{1}}^0, h))$ existe et :
 - * si $(t + e_{\mathbb{1}}^0, \ell) \in \text{Nominal}(\beta_{\mathbb{1}})$ alors $P_{F_{\mathbb{1}}}(e_{\mathbb{1}}^0, p_{\mathbb{1}}^0, t)$ est dérivable et $P_{F_{\mathbb{1}}}(e_{\mathbb{1}}^0, p_{\mathbb{1}}^0, t) = \ell$,
 - * sinon, pour tout $h \geq t$, $P_{F_{\mathbb{1}}}(e_{\mathbb{1}}^0, p_{\mathbb{1}}^0, h) = P_{G_{\mathbb{1}}}(0, \ell - p_{\beta_{\mathbb{1}}}, h - t)$ où $G_{\mathbb{1}}$ est définie comme suit : $\forall u \in \mathbb{R}^+$, $G_{\mathbb{1}}(u) = F_{\mathbb{1}}(u + t)$.

Ceci signifie que pour tout $t \in \mathbb{R}^{+*}$ tel que $(t + e_{\mathbb{1}}^0, \ell) \in \text{Nominal}(\beta_{\mathbb{1}})$, la fonction $P_{F_{\mathbb{1}}}(e_{\mathbb{1}}^0, p_{\mathbb{1}}^0, t)$ est continue. Sinon, elle est discontinue en t avec un saut de $-p_{\beta_{\mathbb{1}}}$;

- Une unique fonction P_{F_P} telle que :
 - $P_{F_P}(\overleftarrow{\mathbb{1}}_P(e_{\mathbb{1}}^0), \overleftarrow{\mathbb{1}}_P(p_{\mathbb{1}}^0), 0) = \overleftarrow{\mathbb{1}}_P(p_{\mathbb{1}}^0) = p_{\mathbb{1}}^0 \times p_{\beta}$ d'après la Définition 25.
Ceci signifie que la valeur initiale de p_P^t est p_{β} fois plus grande que la valeur initiale de $p_{\mathbb{1}}^t$. De plus, on a $\overleftarrow{\mathbb{1}}_P(e^0) = e^0$, ce qui signifie que la valeur initiale de e_P^t est égale à la valeur initiale de $e_{\mathbb{1}}^t$.
 - Pour tout $t \in \mathbb{R}^+$, $\frac{dP_{F_P}(\overleftarrow{\mathbb{1}}_P(e_{\mathbb{1}}^0), \overleftarrow{\mathbb{1}}_P(p_{\mathbb{1}}^0), t)}{dt} = F_P(t) - \gamma_P \times P_{F_P}(\overleftarrow{\mathbb{1}}_P(e_{\mathbb{1}}^0), \overleftarrow{\mathbb{1}}_P(p_{\mathbb{1}}^0), t)$
où $F_P(t) = \sum_{x_P \in \text{Pred}_P(\nabla)} \eta_{x_P}^t(0)$.
D'après 1., on a pour tout t et pour toute synapse $s_P \in \text{Pred}_P(\nabla)$, $\eta_{s_P}^t = \overleftarrow{\mathbb{1}}_P(\eta_{s_{\mathbb{1}}}^t)$ ce qui est égal à $\eta_{s_{\mathbb{1}}}^t \times p_{\beta}$ d'après la Définition 25. De plus, d'après 2., on a pour tout t et pour tout compartiment $c_P \in \text{Pred}_P(\nabla)$, $\eta_{c_P}^t = \overleftarrow{\mathbb{1}}_P(\eta_{c_{\mathbb{1}}}^t)$ ce qui est égal à $\eta_{c_{\mathbb{1}}}^t \times p_{\beta}$ d'après la Définition 25. Par conséquent, $F_P(t) = F_{\mathbb{1}}(t) \times p_{\beta}$; Aussi, d'après la Définition 24, on a $\gamma_P = \gamma_{\mathbb{1}}$. Ceci signifie que pour tout t , les charges entrant dans le soma de P ($F_P(t)$) sont p_{β} fois plus importantes que celles entrant dans le soma de $P_{\mathbb{1}}$ ($F_{\mathbb{1}}(t)$), et puisque le facteur de fuite est le même dans les deux neurones, la relation initiale entre p_P^t et $p_{\mathbb{1}}^t$ est préservée.
 - Pour tout $t \in \mathbb{R}^{+*}$, $\ell = \lim_{h \rightarrow t^-} (P_{F_P}(\overleftarrow{\mathbb{1}}_P(e_{\mathbb{1}}^0), \overleftarrow{\mathbb{1}}_P(p_{\mathbb{1}}^0), h))$ existe et :
 - * si $(t + \overleftarrow{\mathbb{1}}_P(e_{\mathbb{1}}^0), \ell) \in \text{Nominal}(\beta)$ alors $P_{F_P}(\overleftarrow{\mathbb{1}}_P(e_{\mathbb{1}}^0), \overleftarrow{\mathbb{1}}_P(p_{\mathbb{1}}^0), t)$ est dérivable et $P_{F_P}(\overleftarrow{\mathbb{1}}_P(e_{\mathbb{1}}^0), \overleftarrow{\mathbb{1}}_P(p_{\mathbb{1}}^0), t) = \ell$,
 - * sinon, pour tout $h \geq t$, $P_{F_P}(\overleftarrow{\mathbb{1}}_P(e_{\mathbb{1}}^0), \overleftarrow{\mathbb{1}}_P(p_{\mathbb{1}}^0), h) = P_{G_P}(0, \ell - p_{\beta}, h - t)$ où G_P est définie comme suit : $\forall u \in \mathbb{R}^+$, $G_P(u) = F_P(u + t)$.

Ceci signifie que pour tout $t \in \mathbb{R}^{+*}$ tel que $(t + \overleftarrow{\mathbb{1}}_P(e_{\mathbb{1}}^0), \ell) \in \text{Nominal}(\beta)$, la fonction $P_{F_P}(\overleftarrow{\mathbb{1}}_P(e_{\mathbb{1}}^0), \overleftarrow{\mathbb{1}}_P(p_{\mathbb{1}}^0), t)$ est continue. Sinon, elle est discontinue en t avec un saut de $-p_{\beta}$. Or, on sait que $\beta = \beta_{\mathbb{1}} \times p_{\beta}$. Puisque $\overleftarrow{\mathbb{1}}_P(e^0) = e^0$, ceci

signifie que pour tout $t \in \mathbb{R}^{+*}$ où $(t + e_{\mathbb{1}}^0, \ell) \in \text{Nominal}(\beta_{\mathbb{1}} \times p_{\beta})$, la fonction $P_{FP}(\overleftarrow{\mathbb{1}}_P(e_{\mathbb{1}}^0), \overleftarrow{\mathbb{1}}_P(p_{\mathbb{1}}^0), t)$ est continue; sinon elle est discontinue en t avec un saut de $-p_{\beta}$.

On a prouvé que si $t = 0$, p_P^t est p_{β} fois plus grande que $p_{\mathbb{1}}^t$ et $e_P^t = e_{\mathbb{1}}^t$. Ensuite, pour tout $t \in \mathbb{R}^+$ tel que p_P^t et $p_{\mathbb{1}}^t$ sont continues sur $[0, t]$, ces relations sont préservées. De plus, la fonction de borne de P est p_{β} fois plus grande que la fonction de borne de $P_{\mathbb{1}}$ et donc, p_P^t et $p_{\mathbb{1}}^t$ atteindront leur seuil au même instant, rendant les discontinuités des deux fonctions simultanées et préservant ainsi $e_P^t = e_{\mathbb{1}}^t$. La valeur soustraite de p_P est alors p_{β} fois plus grande que la valeur soustraite de $p_{\mathbb{1}}$ puisque $p_{\beta_{\mathbb{1}}} = 1$. Il vient que, pour tout $t \in \mathbb{R}^+$, $p_P^t = p_{\mathbb{1}}^t \times p_{\beta}$ et $e_P^t = e_{\mathbb{1}}^t$.

Enfin, d'après la Définition 25, on a $\overleftarrow{\mathbb{1}}_P(p_{\mathbb{1}}^t) = p_{\mathbb{1}}^t \times p_{\beta}$ ce qui est égal à p_P^t et on a $\overleftarrow{\mathbb{1}}_P(e_{\mathbb{1}}^t) = e_{\mathbb{1}}^t$ ce qui est égal à e_P^t , prouvant 3.

Nous avons donc prouvé, $d_{P,I}(t) = \overleftarrow{\mathbb{1}}_P(d_{P_{\mathbb{1}},I}(t))$ car nous avons prouvé que conformément au signal d'entrée I :

1. la dynamique des synapses de P est identique à la dynamique des synapses de $P_{\mathbb{1}}$ à laquelle est appliquée $\overleftarrow{\mathbb{1}}_P$;
2. la dynamique des compartiments de P est identique à la dynamique des compartiments de $P_{\mathbb{1}}$ à laquelle est appliquée $\overleftarrow{\mathbb{1}}_P$;
3. la dynamique du soma de P est identique à la dynamique du soma de $P_{\mathbb{1}}$ à laquelle est appliquée $\overleftarrow{\mathbb{1}}_P$.

□

À partir du théorème de normalisation, nous pouvons facilement prouver que si l'état initial d'un neurone N est égal à la reconstruction d'état à partir de la dé-normalisation de l'état initial de sa forme normalisée alors cette relation est également vraie pour tout autre état de N (Figure 5.8).

Corollaire 1. [Corollaire du théorème de normalisation]

Soient un neurone N , le neurone normalisé $P_{\mathbb{1}} = \mathbb{1}(N)$ et un état η^0 de $P_{\mathbb{1}}$ en continuité avec un signal d'entrée I . Si $d_{P_{\mathbb{1}},I}$ est la dynamique de $P_{\mathbb{1}}$ (conformément à I avec η^0 comme état initial) et si $d_{N,I}$ est la dynamique de N (conformément à I avec $\overleftarrow{\mathbb{P}}_N(\overleftarrow{\mathbb{1}}_P(\eta^0))$ comme état initial), alors pour tout $t \in \mathbb{R}^+$:

$$d_{N,I}(t) = \overleftarrow{\mathbb{1}}_N(d_{P_{\mathbb{1}},I}(t))$$

Preuve: Par définition, nous devons prouver $d_{N,I}(t) = \overleftarrow{\mathbb{P}}_N(\overleftarrow{\mathbb{1}}_P(d_{P_{\mathbb{1}},I}(t)))$.

- D'après la remarque de la Définition 24, on a : $\mathbb{1}(N) = \mathbb{1}(\mathbb{P}(N)) = P_{\mathbb{1}}$.
On note alors $\mathbb{P}(N) = P$.
- D'après le Théorème 2, on a : $d_{N,I}(t) = \overleftarrow{\mathbb{P}}_N(d_{P,I}(t))$.

5.2. LES NEURONES NORMALISÉS : PROPRIÉTÉS ET LIMITATION

- D'après le Théorème 1, on a : $d_{P,I}(t) = \overleftarrow{\mathbb{1}}_P(d_{P_1,I}(t))$.
Par conséquent, $d_{N,I}(t) = \overleftarrow{\mathbb{P}}_N(\overleftarrow{\mathbb{1}}_P(d_{P_1,I}(t)))$.

□

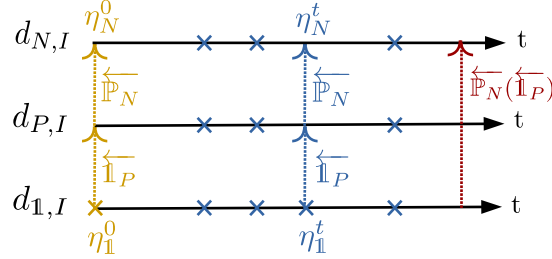


FIGURE 5.8 – Illustration du corollaire du théorème de normalisation. Étant donné un neurone N , sa forme normalisée $\mathbb{1} = \mathbb{1}(N)$ et un signal d'entrée I , si la dynamique de N au temps 0 est la reconstruction d'état à partir de la dé-normalisation de la dynamique de $\mathbb{1}$ au temps 0 (i.e. $d_{N,I}(0) = \eta_N^0 = \overleftarrow{\mathbb{P}}_N(\overleftarrow{\mathbb{1}}_P(d_{\mathbb{1},I}(0))) = \overleftarrow{\mathbb{P}}_N(\overleftarrow{\mathbb{1}}_P(\eta_{\mathbb{1}}^0))$) alors, pour tout temps t , la dynamique de N est égale à la reconstruction d'état à partir de la dé-normalisation de la dynamique de $\mathbb{1}$ au même temps t (i.e. $d_{N,I}(t) = \eta_N^t = \overleftarrow{\mathbb{P}}_N(\overleftarrow{\mathbb{1}}_P(d_{\mathbb{1},I}(t))) = \overleftarrow{\mathbb{P}}_N(\overleftarrow{\mathbb{1}}_P(\eta_{\mathbb{1}}^t))$).

Finalement, grâce au théorème de normalisation et son corollaire, nous prouvons le *théorème d'entrée/sortie* (Théorème 4 ci-après) qui stipule que le processus de normalisation préserve bien la fonction d'entrée/sortie du neurone d'origine : pour un signal d'entrée donné, un neurone et sa forme normalisée possèdent toujours la même sortie.

Théorème 4. [Théorème d'entrée/sortie]

Soient un neurone N , le neurone normalisé $P_{\mathbb{1}} = \mathbb{1}(N)$ et un état η^0 de $P_{\mathbb{1}}$ en continuité avec un signal d'entrée I . Si $O_{P_{\mathbb{1}},I}$ est le signal de sortie de $P_{\mathbb{1}}$ conformément à I avec η^0 comme état initial et si $O_{N,I}$ est le signal de sortie de N conformément à I avec $\overleftarrow{\mathbb{P}}_N(\overleftarrow{\mathbb{1}}_P(\eta^0))$ comme état initial, alors pour tout $t \in \mathbb{R}^+$:

$$O_{N,I}(t) = O_{P_{\mathbb{1}},I}(t)$$

Preuve: On doit prouver $O_{N,I}(t) = O_{P_{\mathbb{1}},I}(t)$. On note $\mathbb{P}(N) = P$. D'après le Corollaire 1, on a $d_{N,I}(t) = \overleftarrow{\mathbb{P}}_N(\overleftarrow{\mathbb{1}}_P(d_{P_{\mathbb{1}},I}(t)))$. Pour tout $t \in \mathbb{R}^+$, on note également :

- $\eta_{\mathbb{1}}^t = d_{P_{\mathbb{1}},I_{\mathbb{1}}}(t) = (\{\eta_{s_{\mathbb{1}}}\}_{s_{\mathbb{1}} \in Sy(P_{\mathbb{1}})}, \{\eta_{c_{\mathbb{1}}}\}_{c_{\mathbb{1}} \in Co(P_{\mathbb{1}})}, (e_{\mathbb{1}}^t, p_{\mathbb{1}}^t))$
- $\eta_P^t = \overleftarrow{\mathbb{1}}_P(\eta_{\mathbb{1}}^t) = (\{\eta_{s_P}\}_{s_P \in Sy(P)}, \{\eta_{c_P}\}_{c_P \in Co(P)}, (e_P^t, p_P^t))$
- $\eta_N^t = \overleftarrow{\mathbb{P}}_N(\eta_P^t) = (\{\eta_{s_N}\}_{s_N \in Sy(N)}, \{\eta_{c_N}\}_{c_N \in Co(N)}, (e_N^t, p_N^t))$

Par définition, la fonction de borne pour le soma de N est p_{β} fois plus importante que la fonction de borne pour le soma de $P_{\mathbb{1}}$, quelque soit le temps écoulé depuis l'émission du dernier spike. Ceci signifie que le potentiel de N (p_N^t) doit être p_{β} fois plus grand que le potentiel de $P_{\mathbb{1}}$ ($p_{\mathbb{1}}^t$) pour atteindre son seuil. D'après la Définition 22, $(e_N^0, p_N^0) = (e_{\mathbb{1}}^0, p_{\mathbb{1}}^0 \times p_{\beta})$ ce qui prouve que p_N^t est effectivement p_{β} fois plus grand que $p_{\mathbb{1}}^t$ en condition initiale (à $t = 0$). D'après le Théorème 3 (étant donné $p_N^0 = p_{\mathbb{1}}^0 \times p_{\beta}$), nous avons prouvé que pour tout $t \in \mathbb{R}^+$, p_N^t est effectivement p_{β} fois plus grand que $p_{\mathbb{1}}^t$. Par conséquent, il vient que

p_N^t et $p_{\mathbb{1}}^t$ atteindront leurs seuils aux mêmes temps, donc $e_N^t = 0$ si et seulement si $e_{\mathbb{1}}^t = 0$, rendant leurs signaux de sortie respectifs $O_{N,I}$ et $O_{P_{\mathbb{1}},I}$ égaux (Définition 17). \square

Une conséquence directe du théorème d'entrée/sortie est exprimée dans son corollaire (Corollaire 2 ci-après). Puisqu'un neurone et sa forme normalisée ont la même fonction d'entrée/sortie, deux neurones ayant la même forme normalisée ont également la même fonction d'entrée/sortie.

Corollaire 2. [Corollaire du théorème d'entrée/sortie]

*Soient N et N' deux neurones tels que $\mathbb{1}(N) = \mathbb{1}(N') = P_{\mathbb{1}}$. Alors, pour tout état initial η^0 de $P_{\mathbb{1}}$ en continuité avec un signal d'entrée I et pour tout $t \in \mathbb{R}^+$, on a :
Le signal de sortie $O_{N,I}(t)$ de N (conformément à I , avec $\overleftarrow{\mathbb{1}}_N(\eta^0)$ comme état initial) est égal au signal de sortie $O_{N',I}(t)$ de N' (conformément à I , avec $\overleftarrow{\mathbb{1}}_{N'}(\eta^0)$ comme état initial).*

Dans cette section, nous avons défini une fonction de normalisation (Définition 24) qui permet de construire la forme normalisée de n'importe quel neurone, tout en préservant sa fonction d'entrée/sortie (Théorème 4).

Un neurone normalisé est un neurone réduit dont le seuil (hors périodes réfractaires) est normalisé à 1 et qui ne provoque aucune atténuation des signaux entre leur réception aux synapses et leur intégration au soma. La perte d'atténuation ainsi que la normalisation du seuil sont compensées par une modification du poids des synapses. L'arborescence dendritique est alors réduite à un ensemble de délais portés par les éventuels compartiments uniques liant les synapses au soma.

5.2.4 Limitation

Tout neurone normalisé représente une classe d'équivalence de structures partageant la même fonction d'entrée/sortie. Cependant, le neurone normalisé n'est pas l'unique représentant canonique d'une fonction d'entrée/sortie. En effet, bien que la forme normalisée soit plus « réduite » que celle définie dans la section précédente, nous pouvons toujours trouver deux neurones normalisés différents ayant la même fonction d'entrée/sortie.

Contre-exemple : Le contre-exemple le plus simple apparaît lorsque l'on considère un neurone normalisé N formé d'une unique synapse $s = \sigma_s$ liée à un soma $\nabla = (\beta, \gamma)$ par un compartiment $c = (1, \Delta_c)$. Si σ_s est très faible et β et γ suffisamment grands, le seuil de N ne sera jamais atteint, quelque soit le signal d'entrée (à cause de la fuite γ et de la période réfractaire du neurone connecté à N) et son signal de sortie sera donc toujours nul. Par conséquent, deux neurones normalisés N et N' différents (avec $\Delta_c \neq \Delta_{c'}$ par exemple), mais ayant ce type de caractéristiques, partagent la même fonction d'entrée/sortie.

Au vu de ce contre-exemple, on pourrait penser qu'ajouter une condition de « fonctionna-

lité » sur les synapses suffirait à garantir que deux neurones normalisés différents aient des fonctions d'entrée/sortie différentes. Seulement, ceci ne semble pas suffisant et il faudrait donc ajouter d'autres conditions restrictives pour prouver cette propriété.

Exemple de contraintes restrictives : *Considérons deux neurones normalisés différents N et N' tels que :*

- *N et N' ont le même soma ;*
- *N et N' ont les mêmes synapses et chaque synapse est suffisamment puissante pour que la réception d'un spike permette d'atteindre le seuil ;*
- *N et N' ont le même état initial.*

Étant données ces conditions restrictives, puisque N et N' sont différents, il existe au moins une synapse s telle que le compartiment liant s au soma diffère entre N et N' . On note alors c le compartiment liant s au soma de N et c' le compartiment liant s au soma de N' . On a donc $\Delta_c \neq \Delta_{c'}$.

Pour prouver que N et N' n'ont pas la même fonction d'entrée/sortie, il suffit de construire un signal d'entrée tel que N et N' ont un signal de sortie différent. On construit donc le signal d'entrée $I_{|s}$ tel qu'il est identiquement nul sauf à la synapse s au temps t_1 . On admet que t_1 est suffisamment grand pour que N et N' ne soient pas en période réfractaire. Puisque la synapse s est identique sur N et N' alors, la trace produite par le spike $I_{|s}^{t_1}$ est la même. Puisque $\Delta_c \neq \Delta_{c'}$ alors, l'intégration de la trace au soma se fera avec un certain décalage dans le temps δ entre les deux neurones. Le seuil de N ne sera donc pas franchi en même temps que celui de N' et les deux neurones normalisés n'auront donc pas la même sortie.

L'exemple précédent permet de prouver que sous des hypothèses particulières, les neurones normalisés peuvent effectivement être les représentants uniques d'une fonction d'entrée/sortie. Néanmoins, on se rend bien compte que les hypothèses utilisées ici sont extrêmement restrictives, faisant perdre tout intérêt à cette propriété. Lorsque l'on tente d'assouplir les conditions, on se heurte à de nombreuses difficultés qui sont parfois très subtiles. En particulier, les paramètres non fixés qui diffèrent entre neurones normalisés peuvent se compenser et donner la même fonction d'entrée/sortie malgré les différences.

Nous ne sommes pas parvenus à définir un jeu d'hypothèses *raisonnables* permettant d'établir l'unicité de la représentation normalisée. C'est donc une question qui reste ouverte. Quoiqu'il en soit, on peut malheureusement affirmer que la forme normalisée ne représente pas *de manière unique* sa classe d'équivalence.

5.3 Portée de nos résultats

Dans le chapitre précédent (Chapitre 4), nous avons défini un modèle de neurone unique adapté à l'étude des assemblées. Il a la particularité d'intégrer des dendrites et ainsi de tenir compte de la morphologie des neurones qui serait un paramètre clé du fonctionnement des assemblées (Chapitre 1, Section, 1.3). Généralement, les modèles qui possèdent cette

caractéristique sont des modèles biophysiques complexes qui ne permettent pas aisément de passer à l'échelle des réseaux (Chapitre 3, Section 3.1). Or, c'est uniquement à cette échelle supérieure que peuvent être étudiées les assemblées. Le modèle que nous proposons est beaucoup plus abstrait, ce qui permet de s'affranchir de cette limitation sans pour autant négliger l'intégration dendritique.

5.3.1 Interprétation biologique

Dans ce chapitre, nous avons établi une procédure permettant de largement simplifier la morphologie des neurones tout en préservant leur comportement d'entrée/sortie. Le neurone, dans sa forme la plus réduite (dite normalisée), ne possède plus de structure arborescente. Les dendrites sont alors représentées par un ensemble de délais attachés aux synapses. Cette représentation met en lumière l'importance de la dimension temporelle apportée par les dendrites. En ce sens, notre cadre de modélisation permet de valider l'approche classique des biologistes dans l'étude des assemblées. En enregistrant des neurones simultanément, les données obtenues sont seulement temporelles et d'après nos résultats, ceci serait suffisant pour prendre en compte l'influence des dendrites dans un contexte de conduction passive..

La forme normalisée représente toute une classe d'équivalence de structures partageant le même comportement, c'est-à-dire que des neurones ayant des morphologies différentes peuvent avoir la même fonction d'entrée/sortie. Ce constat, soulevé par notre approche, suggère un rôle limité de la structure précise des dendrites dans la fonction d'entrée/sortie du neurone. Le rôle fondamental des dendrites dans ce cadre serait finalement d'apporter des délais.

Bien évidemment, ces conclusions sont à considérer sous les hypothèses simplificatrices que nous posons par rapport à la biologie. Les théorèmes liés aux processus de réduction et de normalisation reposent essentiellement sur l'associativité de la somme des charges faite aux embranchements de l'arborescence dendritique. Sommer les charges localement est un choix qui nous semble raisonnable au niveau d'abstraction auquel nous nous trouvons. Une autre hypothèse implicite est l'indépendance des synapses. Effectivement, quelle que soit la position d'une synapse par rapport aux autres, l'effet des spikes qu'elle reçoit sera toujours le même. Or, il existe en biologie le phénomène de *shunting inhibition* [Thorson, 1966] [Basbaum, 2008] qui suggère le contraire : certaines synapses inhibitrices du cerveau (de type $GABA_A$) sont associées à un potentiel d'inversion proche du potentiel de repos ce qui les rend peu efficaces en conditions « normales » mais bien plus puissantes lorsqu'elles se trouvent sur le chemin d'un signal excitateur. Ces synapses, lorsqu'elles sont placées stratégiquement, pourraient donc « mettre leur veto » sur les signaux excitateurs reçus en amont. Ce mécanisme, parmi d'autres, pourrait se révéler pertinent dans l'étude des assemblées, car il pourraient jouer un rôle dans la précision temporelle des spikes [Mann and Paulsen, 2007] [Alvarez-Leefmans and Delpire, 2009].

Notre cadre de modélisation repose sur l'hypothèse de propagation passive du signal, bien

que l'on sache qu'il existe des mécanismes actifs au niveau de différents types de dendrites (Chapitre 1, Section 1.1). Cette hypothèse a pour conséquence de ne pas tenir compte des non-linéarités apportées par ces mécanismes, mais prendre en compte de telles propriétés nuirait au passage à l'échelle des réseaux de neurones biologiques. Naturellement, puisque nous modélisons le soma à la manière du modèle Leaky Integrate-and-Fire, nous héritons également de certaines hypothèses et limites sous-jacentes à ce modèle [Izhikevich, 2004].

Néanmoins, comme on l'a vu, l'apport d'une modélisation de l'intégration dendritique est un préalable à l'étude des assemblées de neurones et notre formalisme en constitue une première étude systématique.

Puisque différentes structures peuvent être associées à une même fonction, nos résultats suggèrent que la forme exacte des dendrites n'est pas le paramètre déterminant de la fonction neuronale. Il suffirait finalement que l'arborisation du neurone fasse partie de la classe d'équivalence correspondant à la fonction qu'il doit remplir. Pourtant, certains types de neurones ont une forme caractéristique [Mel, 1994]. La structure arborescente des neurones pourrait donc être dictée préférentiellement par d'autres critères comme des contraintes spatiales ou énergétiques. La morphologie privilégiée pourrait permettre d'augmenter la surface de contact tout en limitant les redondances, et en optimisant la quantité de membrane [Mitchison, 1991] [Chklovskii et al., 2002] [Cuntz et al., 2010] [Budd and Kisvárdy, 2012] [Puppo et al., 2018].

Par ailleurs, la mise en évidence des classes d'équivalence suggère une classification des neurones différente de l'habituelle classification morphologique (Chapitre 1, Section 1.1). Des neurones qui semblent *a priori* différents sur le plan morphologique (dendrites), mais qui jouent en fait le même rôle, seraient alors rapprochés dans la classification. Nos résultats suggèrent que c'est la fonction d'entrée/sortie équivalente qui serait fondamentale, alors que la structure très précise des dendrites serait soumise à des contraintes locales (mécaniques, *etc.*), plus épiphénoménales.

Finalement, le cadre de modélisation que nous proposons dans cette thèse est adapté à une mise à l'échelle et à l'étude des assemblées, mais également pour l'étude d'autres phénomènes pour lesquels on s'intéresse aux aspects temporels. À l'échelle d'un neurone unique, nos résultats soulèvent des problématiques relatives aux contraintes qui influencent la structure des neurones. Nos résultats pourraient également orienter les recherches en neurosciences en mettant en exergue les différences entre la réalité biologique et notre neurone idéalisé, pointant ainsi des fonctions biologiques plus fines.

5.3.2 Apport à l'informatique bio-inspirée

La procédure de normalisation que nous présentons dans ce chapitre aboutit à une importante simplification de notre modèle. La représentation de l'arborescence dendritique en un ensemble de délais permet de s'affranchir de la représentation explicite sous forme d'arbre spatial. D'un point de vue algorithmique, la complexité de la dynamique est également ré-

duite puisqu'il n'est plus nécessaire de calculer le glissement pas à pas des signaux dans l'arborescence.

Nous avons prouvé que la forme normalisée représente toute une classe d'équivalence de structures partageant la même fonction d'entrée/sortie. Ainsi, étudier un neurone normalisé revient à explorer toute sa classe d'équivalence, ce qui s'avère particulièrement intéressant en simulation.

Par ailleurs, nous démontrons dans ces travaux que les délais (avec des propriétés synaptiques et somatiques adaptées) capturent l'impact de l'intégration des signaux dans l'arborescence dendritique. Ajouter des délais à des modèles de neurones ponctuels est donc un moyen simple et efficace de représenter une intégration dendritique abstraite. Sous l'angle de vue inverse, les modèles qui intègrent des délais, généralement pour représenter la transmission axonale ou synaptique [Usher et al., 1993] [Gerstner et al., 1996] [Izhikevich, 2006] [Datadien, 2010], peuvent finalement être utilisés pour représenter plus de choses que ce pour quoi ils ont été initialement conçus.

Enfin, le cadre de modélisation que nous proposons possède des propriétés algébriques très intéressantes qui seront mises en lumière dans le chapitre suivant. Cette approche nous permet de raisonner rigoureusement sur les liens entre neurones qui seront représentés comme des objets mathématiques bien définis.

CHAPITRE 6

UNE VISION PUREMENT ALGÈBRIQUE

Dans ce chapitre, nous proposons une vision algébrique du cadre de modélisation défini dans le Chapitre 4. La théorie des catégories nous permet d'étendre les résultats présentés dans le Chapitre 5. Nous avons précédemment prouvé que pour tout neurone, il existe une unique forme réduite, ainsi qu'une unique forme normalisée de ce neurone, préservant sa fonction d'entrée/sortie. Les processus de réduction et de normalisation éliminent en grande partie la structure arborescente des neurones. Ici, on définit un « continuum » entre le neurone et sa forme normalisée, chacun des neurones intermédiaires ayant une arborescence dendritique plus ou moins élaborée, et partageant tous la même fonction d'entrée/sortie. Ceci permet de caractériser plus finement la composition des classes d'équivalence mises en évidence dans le chapitre précédent.

Ce chapitre a également pour objectif de *prendre du recul* sur les résultats précédemment obtenus, pour mieux comprendre l'intégration dendritique. Le processus de réduction, présenté dans le Chapitre 5, consiste globalement à « retarder » une multitude de sommes faites pas à pas dans les branches de l'arborescence dendritique en une unique somme faite au niveau du soma. La préservation de la fonction d'entrée/sortie des neurones par ce processus semble assez évidente à première vue. Effectivement, on pourrait facilement penser que ce résultat est une simple conséquence des propriétés d'associativité et de commutativité de l'addition. Cependant, ce serait oublier la dynamique de glissement des charges le long de l'arborescence : c'est ce qui explique la longueur des démonstrations permettant d'aboutir à ce résultat fondamental. Il faut en fait montrer la compatibilité de ces propriétés d'associativité et de commutativité avec un glissement dans le temps selon une structure arborescente, et c'est ce qui constitue l'essentiel de la preuve : par ailleurs, il faut montrer la compatibilité entre le glissement dans le temps du signal d'entrée discret, le glissement du signal continu produit par les synapses à partir du signal d'entrée discret, le glissement arborescent avec les sommes faites dans l'arborescence dendritique et la somme globale faite au soma. Dans ce chapitre, nous mettons en évidence ces compatibilités successives grâce à la notion de

morphisme de foncteurs.

6.1 La catégorie des neurones

6.1.1 Définitions préliminaires

De la même manière que dans le Chapitre 4 (Définition 7), on définit dans ce chapitre, un neurone N comme un arbre étiqueté dont l'ensemble des nœuds $Nodes(N)$ est constitué de l'union des sous-ensembles $Sy(N)$, $Co(N)$ et $\{\nabla\}$. $Sy(N)$ est l'ensemble des synapses de N , $Co(N)$ est l'ensemble de ses compartiments et ∇ est son soma. Le soma et les compartiments sont respectivement définis d'après les Définitions 4 et 6 du Chapitre 4. Seules les définitions des synapses et du soma sont enrichies d'une convention supplémentaire : en plus d'être caractérisée par une fonction synaptique, chaque synapse s est également munie d'un facteur d'atténuation $\alpha_s = 1$ (aucune atténuation) et d'un délai de traversée $\Delta_s = 0$ artificiels. Il en est de même pour le soma ∇ . Les définitions des synapses et du soma sont désormais cohérentes avec celle des compartiments, ce qui permet d'homogénéiser les notations et ainsi de simplifier nos définitions.

Notation 3. [Synapses et soma]

Par convention, étant donnée une synapse s , on dira qu'elle porte un délai nul $\Delta_s = 0$ et une non atténuation $\alpha_s = 1$. De même, étant donné un soma ∇ , on dira qu'il porte un délai $\Delta_\nabla = 0$ et une atténuation $\alpha_\nabla = 1$.

Dans le Chapitre 5, nous avons défini un chemin $Path_N(x \rightarrow y)$ entre deux nœuds x et y dans un neurone N . Si ce chemin existe, il s'agit de la séquence de compartiments qui lie x et y , en direction du soma de N . Ce chemin inclut alors les deux nœuds x et y . Ici, nous modifions légèrement cette définition, pour exclure x du chemin de x à y dans N , ce qui simplifie les notations et preuves à venir, en particulier car la concaténation ne compte qu'une fois le nœud de jonction. Pour distinguer cette nouvelle définition de chemin de celle du Chapitre 5, on le note dorénavant $]x_{\overline{N}} \rightarrow y]$:

Définition 26. [Chemin dans un neurone - Nouvelle définition]

Soient un neurone N et deux nœuds $x, y \in Nodes(N)$. Soit $S = c_0, \dots, c_n$ l'unique suite de nœuds liant x au soma ∇ de N où $c_0 = x$ et $c_n = \nabla$. Le chemin de x à y dans N , noté $]x_{\overline{N}} \rightarrow y]$, est défini comme suit.

- *S'il existe i tel que $y = c_i$ alors $]x_{\overline{N}} \rightarrow y] = c_1, \dots, c_i$ (et $]x_{\overline{N}} \rightarrow y] = \emptyset$ si $x = y$).*
- *Sinon, $]x_{\overline{N}} \rightarrow y]$ est indéfini.*

On peut alors définir de nouveau les concepts clés d'atténuation et de délai cumulés entre deux nœuds x et y d'un neurone N . Ils caractérisent respectivement l'atténuation globale

appliquée aux charges traversant les branches de l'arbre dendritique de N , entre la sortie du nœud x et la sortie du nœud y , et le temps de traversée de ces branches. Si $[x_{\overline{N}} \rightarrow y]$ existe, alors :

- le *délat cumulé* de x à y vaut $\Sigma_N^{x \rightarrow y} = \sum_{c \in [x_{\overline{N}} \rightarrow y]} \Delta_c$;
- l'*atténuation cumulée* de x à y vaut $\Pi_N^{x \rightarrow y} = \prod_{c \in [x_{\overline{N}} \rightarrow y]} \alpha_c$.

6.1.2 Morphismes de neurones

Dans le Chapitre 5, nous avons prouvé que le processus de réduction préserve la fonction d'entrée/sortie des neurones. Rappelons que le processus de réduction consiste à lier chaque synapse au soma d'un neurone par, au plus, un unique compartiment, en ne changeant ni les paramètres des synapses ni ceux du soma. La structure arborescente des dendrites disparaît alors. La préservation de la fonction d'entrée/sortie, à l'issue de ce processus, provient essentiellement de la préservation de l'atténuation et du délat cumulé entre chaque synapse et le soma. Nous avons également montré, avec la normalisation, qu'il est aussi possible de préserver la fonction d'entrée/sortie des neurones en jouant sur les paramètres des synapses et du soma. Finalement, dans le chapitre précédent, nous avons montré la préservation de la fonction d'entrée/sortie des neurones avec la suppression de l'arborescence dendritique. Ici, nous allons montrer qu'il existe en fait un « continuum » de neurones entre un neurone donné et sa forme normalisée, partageant la même fonction d'entrée/sortie. La structure arborescente est alors plus ou moins élaborée. Ceci passe par la définition de la *catégorie des neurones*, dont les *objets minimaux* sont les neurones normalisés (cf. Chapitre 2, Section 2.3). Les *morphismes* de neurones formalisent les liens entre les neurones d'une même classe d'entrée/sortie, permettant de les comparer de manière élégante :

Pour qu'il existe un morphisme d'un neurone N vers un neurone N' , il faut tout d'abord que N et N' aient le même ensemble de synapses (plus précisément les mêmes symboles). Les (symboles de) compartiments sont également préservés par le morphisme. Cependant, il peut exister des compartiments supplémentaires dans N' . Les morphismes vont ainsi dans le sens de l'éloignement de la forme normalisée : N' , s'il est différent de N , sera donc d'arborescence plus riche, plus éloignée de sa forme normalisée. Dans l'objectif de conserver la fonction d'entrée/sortie des neurones, le délat cumulé entre deux nœuds x et y (incluant x) est préservé par le morphisme. Par ailleurs, le seuil du soma de N' (quelque soit son état) est proportionnel et supérieur au seuil du soma de N , de manière à ce que le seuil minimal soit celui de la forme normalisée (égal à 1 hors période réfractaire). De plus, le soma de N' a le même facteur de fuite que celui du soma de N . Similairement, les fonctions synaptiques des synapses de N' peuvent être différentes de celles des synapses de N mais elles doivent être proportionnelles et supérieures à celles-ci. Enfin, si la fonction synaptique d'une synapse de N' est q_μ^s fois plus grande que celle de la synapse correspondante dans N et que le seuil de N' est q_μ^∇ fois plus grand que celui du soma de N alors, l'atténuation

cumulée de la synapse au soma de N' doit être adaptée pour que la charge à l'entrée du soma de N' soit q_μ^∇ fois plus grande qu'à l'entrée du soma de N .

Définition 27. [Catégorie des neurones et morphismes]

Un morphisme de neurones $\mu : N \rightarrow N'$ est une application qui préserve les symboles de $Nodes(N)$ et leur type (synapse, compartiment ou soma) dans $Nodes(N')$, telle que :

1. $\mu(Sy(N)) = Sy(N')$ (i.e. μ est surjectif sur les synapses) ;
2. pour tout nœud $x \in Nodes(N)$, si y est le successeur de x dans N alors il existe un chemin $[\mu(x) \xrightarrow{N'} \mu(y)]$;
3. pour tout nœud $x \in Nodes(N)$, $\Delta_x + \sum_N^{x \rightarrow \nabla} = \Delta_{\mu(x)} + \sum_{N'}^{\mu(x) \rightarrow \mu(\nabla)}$;
4. le quotient $q_\mu^\nabla = \frac{\beta'}{\beta}$ est une fonction constante appartenant à $[1, +\infty[$ (où β et β' sont respectivement les fonctions de borne du soma ∇ de N et du soma $\mu(\nabla)$ de N') et $\gamma' = \gamma$ (où γ et γ' sont respectivement les facteurs de fuite du soma ∇ de N et du soma $\mu(\nabla)$ de N') ;
5. pour toute synapse $s \in Sy(N)$, $q_\mu^s = \frac{\sigma'_s}{\sigma_s}$ est une fonction constante appartenant à $[1, +\infty[$ (où σ_s et σ'_s sont respectivement les fonctions synaptiques des synapses s et $\mu(s)$) ;
6. pour toute synapse $s \in Sy(N)$, $q_\mu^s \times \prod_{N'}^{\mu(s) \rightarrow \mu(\nabla)} = q_\mu^\nabla \times \prod_N^{s \rightarrow \nabla}$.

On note \mathcal{N} la catégorie des neurones, dont les objets sont les neurones et les morphismes sont les morphismes de neurones.

D'après la Définition 27, si $\mu : N \rightarrow N'$ est un morphisme alors :

- l'ensemble des symboles de N est inclus dans l'ensemble des symboles de N' . Par conséquent, on s'autorisera parfois les notations $[x \xrightarrow{N'} y]$, $\sum_{N'}^{x \rightarrow y}$ et $\prod_{N'}^{x \rightarrow y}$ au lieu de $[\mu(x) \xrightarrow{N'} \mu(y)]$, $\sum_{N'}^{\mu(x) \rightarrow \mu(y)}$ et $\prod_{N'}^{\mu(x) \rightarrow \mu(y)}$;
- on déduit que pour tout chemin non vide entre deux nœuds x et y dans N , $\Delta_x + \sum_N^{x \rightarrow y} = \Delta_{\mu(x)} + \sum_{N'}^{\mu(x) \rightarrow \mu(y)}$; ceci résulte de la propriété générale $\sum_N^{x \rightarrow z} = \sum_N^{x \rightarrow y} + \sum_N^{y \rightarrow z}$ avec $z = \nabla$;
- pour tout nœud $x \in Nodes(N)$, $\Delta_{\mu(x)} \leq \Delta_x$;
- μ préserve le successeur de toutes les synapses, car il ne peut y avoir aucun ajout de compartiment entre une synapse et son successeur puisque le délai Δ_c est toujours strictement positif ;
- on déduit que $e'_\beta = e_\beta$, $p'_\beta = q_\mu^\nabla \times p_\beta$, $\hat{p}'_\beta = q_\mu^\nabla \times \hat{p}_\beta$ et pour toute synapse $s \in Sy(N)$, $\hat{\tau}'_s = \hat{\tau}_s$, $\tilde{\tau}'_s = \tilde{\tau}_s$ et $w'_s = q_\mu^s \times w_s$.

Globalement, ces conditions permettent de préserver le rapport entre le potentiel au soma et le seuil.

L'exemple le plus simple de morphisme est celui qui va de la forme normalisée d'un neurone quelconque vers ce neurone, en préservant sa fonction d'entrée/sortie. Cependant, il en

existe bien d'autres, laissant place à tout un continuum de transformations. Nous décrivons ci-après des exemples typiques de morphismes.

Redistribution de compartiments

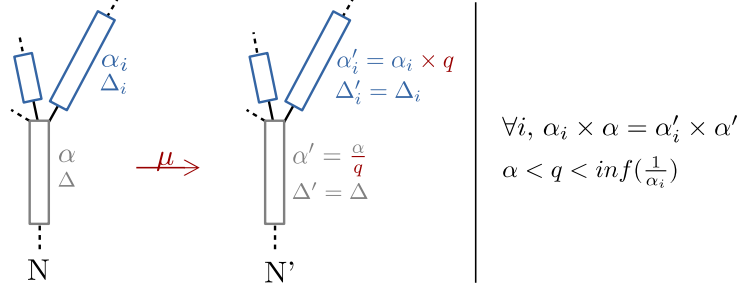


FIGURE 6.1 – Morphisme caractérisé par la redistribution des atténuations sur les compartiments.

Si c est un compartiment de N dont les prédécesseurs sont tous des compartiments, alors le neurone N' peut être obtenu en multipliant par un coefficient q l'atténuation α_i de tous les prédécesseurs c_i de c et en divisant par ce même coefficient q l'atténuation α de c , en gardant tous les autres paramètres inchangés. Ainsi $\alpha'_i \times \alpha' = \alpha_i \times \alpha$. Le coefficient q doit être compris entre α et $\inf(\frac{1}{\alpha_i})$, de sorte que les atténuations $\alpha'_i = q \times \alpha_i$ et $\alpha' = q \times \alpha$ restent inférieures ou égales à 1 (Définition 4, Chapitre 4). L'identité sur les symboles définit alors un morphisme de N à N' . Ce morphisme est un *isomorphisme* (Chapitre 2, Section 2.3). Il existe effectivement un morphisme inverse de N' vers N en utilisant le coefficient multiplicatif $\frac{1}{q}$, qui est toujours compris entre $\frac{\alpha}{q}$ et $\inf(\frac{1}{q \times \alpha_i})$.

Factorisation de compartiments avec ajout (type 1)

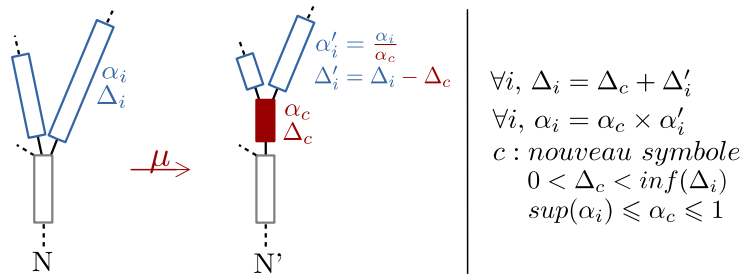


FIGURE 6.2 – Morphisme caractérisé par la factorisation de compartiments, avec ajout d'un nouveau compartiment.

Si c_1, \dots, c_n sont des compartiments ayant un même successeur x dans N , alors le neurone N' peut être obtenu en ajoutant un nouveau compartiment c entre les c_i et leur successeur commun (le symbole c doit être absent du neurone N). L'ensemble des c_i peut aussi bien être l'ensemble $Pred_N(x)$ qu'un sous-ensemble strict de celui-ci. Les paramètres du compartiment c doivent alors satisfaire : $\sup(\alpha_i) \leq \alpha_c \leq 1$ et $0 < \Delta_c < \inf(\Delta_i)$ de sorte que,

pour tout i , $\alpha'_i = \frac{\alpha_i}{\alpha_c}$ et $\Delta'_i = \Delta_i - \Delta_c$. L'inclusion des symboles définit un morphisme de N vers N' . Ce morphisme n'est pas un isomorphisme, car il n'y a pas d'image inverse possible pour le nouveau compartiment c dans N .

Factorisation de compartiments sans ajout (type 2)

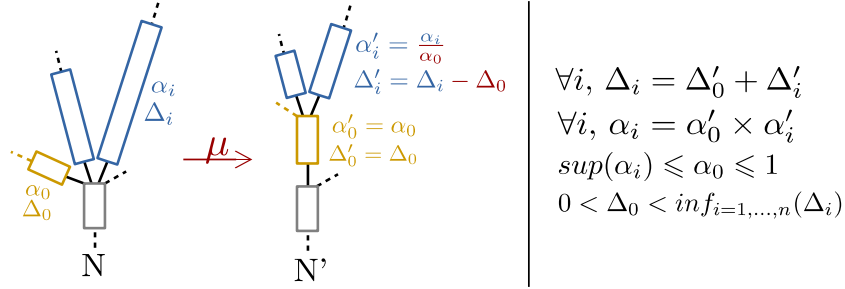


FIGURE 6.3 – Morphisme caractérisé par la factorisation de compartiments, sans ajout de compartiment.

Si c_0, \dots, c_n sont des compartiments ayant un même successeur x dans N tel que $\sup(\alpha_i) \leq \alpha_0 \leq 1$ et $0 < \Delta_0 < \inf_{i=1, \dots, n}(\Delta_i)$, alors le neurone N' peut être obtenu en connectant les compartiments c_1 à c_n au compartiment c_0 . Les paramètres des c_i pour $i = 1, \dots, n$ doivent alors satisfaire : $\alpha'_i = \frac{\alpha_i}{\alpha_0}$ et $\Delta'_i = \Delta_i - \Delta_0$. L'inclusion des symboles définit un morphisme de N vers N' . Ce morphisme n'est pas un isomorphisme, car, par exemple, c_0 est le successeur de c_1 dans N' et le chemin entre c_1 et c_0 n'est pas défini dans N .

Renforcement des synapses

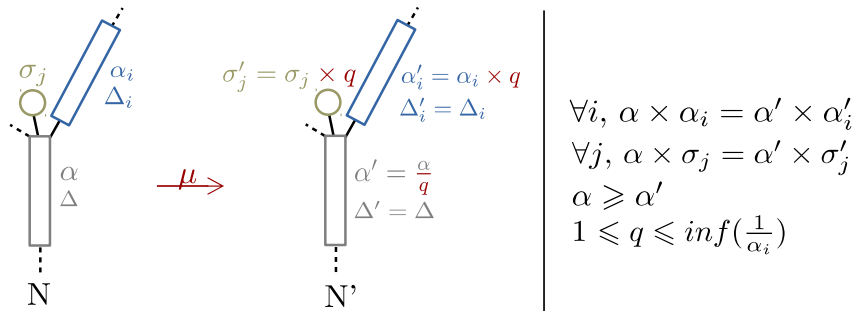


FIGURE 6.4 – Morphisme caractérisé par le renforcement des synapses.

Ce cas est similaire à la *redistribution de compartiments*. Cependant, il s'applique lorsque les prédécesseurs d'un compartiment c de N sont composés non seulement de compartiments, c_i mais aussi de synapses s_j . Ici, le coefficient multiplicatif q doit satisfaire $1 \leq q \leq \inf(\frac{1}{\alpha_i})$ et s'applique aux atténuations des c_i mais aussi aux fonctions synaptiques des s_j . Si $q = 1$, le morphisme est l'identité, sinon ce morphisme n'est pas un isomorphisme, car $\frac{1}{q} < 1$, ce qui contredit la Définition 27.

Renforcement du soma

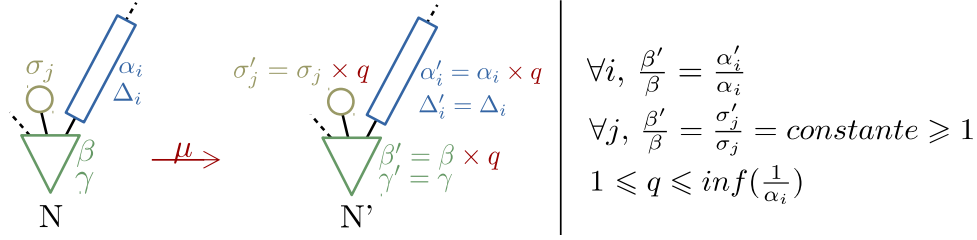


FIGURE 6.5 – Morphisme caractérisé par le renforcement du soma.

Ce cas est l'adaptation du *renforcement des synapses* au cas des prédécesseurs du soma, et non plus des prédécesseurs d'un compartiment. Soit q le coefficient multiplicatif satisfaisant $1 \leq q \leq \inf(\frac{1}{\alpha_i})$ et $\beta' = \beta \times q$. Le neurone N' est alors défini comme pour le cas précédent, hormis pour la fonction de borne du soma. Si $q = 1$, le morphisme est l'identité, sinon ce morphisme n'est pas un isomorphisme, car $\frac{1}{q} < 1$, ce qui contredit la Définition 27.

Nous verrons par la suite que ces exemples de morphismes suffisent à générer tous les morphismes d'un neurone normalisé vers les autres neurones de sa classe d'équivalence.

6.1.3 Propriétés des morphismes

Puisque les morphismes préservent les symboles, on a $|\text{Hom}_{\mathcal{N}}(N, N')| \leq 1$ pour tout couple de neurones N et N' fixés. Par conséquent, tous les morphismes sont des *monomorphismes* et des *épimorphismes*, même si l'application sous-jacente n'est pas surjective (puisque tout nœud de N' n'a pas nécessairement de nœud correspondant dans N).

Remarque: Bien que l'on se place dans la théorie des catégories, on peut néanmoins utiliser le terme de *surjection*, car la catégorie des neurones est une catégorie concrète : les applications sous-jacentes aux morphismes concernent des ensembles de symboles.

La catégorie des neurones est donc une catégorie particulière dans laquelle les notions de monomorphisme et d'épimorphisme n'ont pas d'intérêt. De plus, elles ne sont pas liées au fait que les applications sous-jacentes soient injectives ou surjectives. Néanmoins, puisque le monomorphisme et l'épimorphisme n'impliquent pas toujours l'isomorphisme, il est judicieux de s'intéresser aux isomorphismes. En effet, comme nous l'avons vu dans les exemples précédents (notamment pour le morphisme par renforcement du soma avec $q \neq 1$), tous les morphismes ne sont pas des isomorphismes.

Pour rappel, un morphisme $\mu : N \rightarrow N'$ est un isomorphisme si et seulement s'il existe un morphisme $\mu^{-1} : N' \rightarrow N$, inverse de μ à droite et à gauche : $\mu^{-1} \circ \mu = \text{Id}_N$ et $\mu \circ \mu^{-1} = \text{Id}_{N'}$. Notons que deux neurones isomorphes ne sont pas nécessairement égaux, mais ils partagent au moins le même arbre sous-jacent (Définition 27).

La proposition ci-après établit que les isomorphismes sont les morphismes surjectifs (c'est-à-dire, qui conservent exactement les mêmes symboles) et qui préservent les paramètres

des synapses et du soma. En effet, si un morphisme $\mu : N \rightarrow N'$ n'est pas surjectif, de nouveaux compartiments peuvent par exemple être créés dans N' , rendant l'inversion impossible puisque les nouveaux compartiments de N' n'ont pas d'image dans N . De plus, si les fonctions synaptiques des synapses ou la fonction de borne du soma de N' sont différentes de celles de N , il n'est notamment pas garanti que le coefficient multiplicatif $\frac{1}{q}$ respecte les conditions requises par les morphismes (avec q le coefficient multiplicatif impliqué dans le morphisme μ).

Proposition 1. [Isomorphismes de \mathcal{N}]

Les isomorphismes de la catégorie \mathcal{N} sont les morphismes surjectifs tels que, avec les notations de la Définition 27, $q_\mu^\nabla = q_\mu^s = 1$.

Preuve: Si un morphisme $\mu : N \rightarrow N'$ est un isomorphisme alors les neurones N et N' ont exactement le même ensemble de nœuds. L'application sous-jacente est donc bijective. Par conséquent, μ est surjectif (Chapitre 2, Section 2.1). Puisque μ est un isomorphisme, $q_{\mu^{-1}}^\nabla = \frac{1}{q_\mu^\nabla}$ et $q_{\mu^{-1}}^s = \frac{1}{q_\mu^s}$ et donc $q_\mu^\nabla \times q_{\mu^{-1}}^\nabla = q_\mu^s \times q_{\mu^{-1}}^s = 1$. De plus, ces quatre quotients sont, par définition, supérieurs ou égaux à 1. Par conséquent, pour satisfaire l'équation précédente, ils doivent être égaux à 1. Réciproquement, tous les morphismes sont injectifs et donc ceux qui sont surjectifs sont bijectifs. Il est alors facile de prouver que lorsque $q_\mu^\nabla = q_\mu^s = 1$, l'application μ^{-1} satisfait les conditions de la Définition 27 et donc, que μ^{-1} est un morphisme de neurones. \square

6.1.4 Propriétés des objets

La catégorie \mathcal{N} ne contient ni objet *terminal*, ni objet *initial*. Effectivement, il n'existe pas de neurone vers lequel vont des morphismes provenant chacun des neurones de \mathcal{N} , ni de neurone duquel partent des morphismes vers chaque neurone de \mathcal{N} . Cependant, il existe des objets *minimaux* (Théorème 5 ci-après), qui sont particulièrement intéressants.

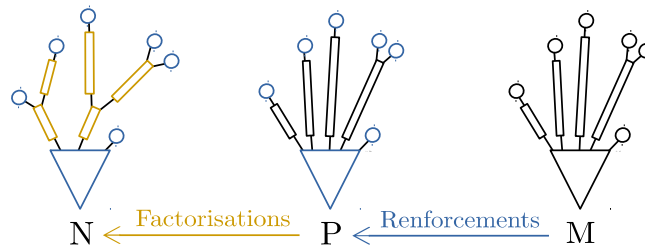


FIGURE 6.6 – Le neurone M est un objet minimal de la catégorie \mathcal{N} . Pour tout neurone N , il existe un neurone minimal tel que N peut être obtenu par composition de morphismes par renforcement des synapses et du soma et par factorisations de compartiments successives.

Pour tout neurone N de \mathcal{N} , il existe un neurone minimal M à partir duquel il est possible d'obtenir N par morphisme. De plus, il n'existe aucun autre morphisme vers M que l'iden-

tité. Les objets minimaux correspondent en fait aux neurones normalisés (Chapitre 5). Les neurones minimaux ont ainsi des dendrites sans structure arborescente, et ne provoquant aucune atténuation. De plus, leur seuil de leur soma vaut 1 (hors périodes réfractaires).

Théorème 5. [Théorème des neurones minimaux]

Un neurone M est dit minimal si la hauteur de son arbre sous-jacent vaut 1 ou 2 et si :

- $p_\beta = 1$ où β est la fonction de borne du soma de M ;
- pour tout $c \in Co(M)$, $\alpha_c = 1$.

Les neurones minimaux sont les objets minimaux de la catégorie \mathcal{N} , ce qui signifie que pour tout neurone N :

1. il existe un neurone minimal M et un morphisme $\mu : M \rightarrow N$;
2. à l'inverse, si $\mu : N \rightarrow M$ est un morphisme où M est un neurone minimal, alors $\mu = Id_M$.

Nous pouvons donc considérer, pour tout neurone N , le neurone minimal de N .

Preuve:

1. Étant donné un neurone N , prenons le neurone P obtenu en connectant directement chaque synapse $s \in Sy(N)$ à ∇ , si c'est déjà le cas dans N , ou via un unique compartiment portant le même symbole c_s que le successeur de s dans N , avec $\Delta_{c_s} = \Sigma_N^{s \rightarrow \nabla}$ et $\alpha_{c_s} = \Pi_N^{s \rightarrow \nabla}$. De plus, P garde le même soma que N ($\beta_P = \beta_N$ et $\gamma_P = \gamma_N$). On remarque que P est la forme réduite de N , au sens du Chapitre 5. De toute évidence, il y a un morphisme de P vers N composé de factorisations successives. Ceci se prouve par induction sur la structure arborescente de N du soma vers les synapses, c'est-à-dire en regardant chaque sous-arbre de N , en commençant ceux qui sont liés à ∇ :

- Si tous les $Pred_N(\nabla)$ sont soit des synapses, soit des compartiments c tels que $CoPred_N(c) = \emptyset$, alors $N = P$.
- Sinon, pour tout compartiment $c \in Pred_N(\nabla)$ tel que $CoPred_N(c) \neq \emptyset$:
 - si c est un « tronc commun » tel que $SyPred_N(c) = \emptyset$, alors on peut trouver un neurone N' qui permet d'obtenir N par factorisation de compartiments de type 1 ;
 - si c est un « tronc commun » tel que $SyPred_N(c) \neq \emptyset$, alors on peut trouver un neurone N' qui permet d'obtenir N par factorisation de compartiments de type 2.

Quel que soit son type, la factorisation réduit strictement la somme des hauteurs des synapses d'un neurone. La suite (strictement décroissante) des hauteurs obtenues par les factorisations successives n'est donc pas infinie. Le processus de factorisations s'arrête lorsque tous les $Pred_N(\nabla)$ sont soit des synapses, soit des compartiments c tels que $CoPred_N(c) = \emptyset$ car sinon on pourrait encore appliquer une factorisation. Le

CHAPITRE 6. UNE VISION PUREMENT ALGÈBRIQUE

neurone obtenu par la dernière factorisation est alors égal à P . Ceci prouve qu'il existe bien un morphisme de P vers N par composition de factorisations.

Dès lors, prenons le neurone M obtenu à partir du neurone P , en remplaçant la fonction de borne β_P par $\beta_M = \frac{\beta_P}{p_\beta}$, α_c par 1 pour tous les compartiments et σ_s par $\frac{\sigma_s \times \alpha_{cs}}{p_\beta}$ pour toutes les synapses. On remarque que M est la forme normalisée de P et donc de N , au sens du Chapitre 5. L'identité sur les symboles définit un morphisme de neurones de M vers P . Il peut être décomposé en morphismes par renforcement des synapses (de profondeur 2) et du soma. Pour tout neurone N , il existe donc un morphisme de P vers N et de M vers P . Par conséquent, il existe bien $\mu : M \rightarrow N$ par composition de ces morphismes (Figure 6.6).

2. Si N est un neurone tel qu'il existe un morphisme $\mu : N \rightarrow M$ alors il est nécessairement surjectif (puisque M contient, au plus, un compartiment entre chaque synapse et le soma). Puisque M est un neurone minimal, $p_\beta = 1$ et par définition, pour N , $p_\beta \geq 1$. Ceci implique $q_\mu^\nabla = 1$ (Définition 27). De plus, puisque M est un neurone minimal, l'atténuation vaut 1 entre chacune de ses synapses et son soma. L'atténuation portée par chaque compartiment de N doit donc être égale à q_μ^s (Définition 27), ce qui implique, $q_\mu^s = 1$ car par définition, les atténuations sont toujours comprises entre 0 et 1 et $q_\mu^s \geq 1$ (Définition 27). D'après la Proposition 1, N et M sont donc isomorphes. De plus, nous savons que M et N partagent les mêmes fonctions synaptiques, les mêmes atténuations, les mêmes délais (Définition 27), le même β , le même γ (Définition 27). Par conséquent $N = M$.

□

Remarque: D'après la preuve du Théorème 5, on établit que la factorisation des compartiments, le renforcement des synapses et le renforcement du soma permettent de générer, par composition, tous les morphismes partant d'un neurone minimal.

Il découle du Théorème 5 que s'il existe un morphisme entre deux neurones, alors ces deux neurones ont le même neurone minimal. Il s'agit là d'une généralisation du cas particulier où l'un des neurones est la forme réduite de l'autre (Remarque de la Définition 24, Chapitre 5).

Corollaire 3. [Corollaire du théorème des neurones minimaux]

Si $\mu : N \rightarrow N'$ est un morphisme de neurones alors N et N' ont le même neurone minimal.

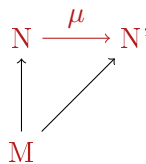


FIGURE 6.7 – Illustration du corollaire 3.

Les sections qui suivent montrent que les morphismes de neurones définis ici sont cohérents avec la dynamique et permettent bien de conserver la fonction d'entrée/sortie des neurones.

6.2 Les catégories de signaux d'entrée

Dans cette section, nous définissons d'une part la catégorie des *signaux d'entrée discrets* et d'autre part la catégorie des *signaux d'entrée continus*. Un signal d'entrée discret représente les spikes reçus par un neurone au cours du temps et un signal d'entrée continu correspond à la trace de ces spikes « à la sortie » des synapses du neurone.

6.2.1 Signaux d'entrée discrets

Dans le Chapitre 4 (Définitions 10 et 11), nous avons défini un signal d'entrée (discret) pour un neurone N comme une famille de segments $\{I_s\}_{s \in \text{Sy}(N)}$ où pour toute synapse s et pour tout $t \in \mathbb{R}$, si s reçoit un spike au temps t alors $I_s(t)$ vaut 1, sinon $I_s(t)$ vaut 0. Dans ce chapitre, afin d'obtenir des propriétés algébriques intéressantes, nous allons définir des *sommes* de signaux d'entrée discrets. Si on somme des signaux d'entrée discrets (suivant la Définition 1 du Chapitre 2), nous souhaitons naturellement obtenir un signal d'entrée discret. Or, si par exemple on somme deux signaux d'entrée discrets identiques, $I_s(t)$ pourra être supérieur à 1 si un spike est reçu par s au temps t . Par conséquent, nous devons élargir la définition proposée dans le Chapitre 4 en définissant $I_s : \mathbb{R} \rightarrow \mathbb{N}$ au lieu de $I_s : \mathbb{R} \rightarrow \{0, 1\}$ seulement. On qualifiera de *plausible*, un signal d'entrée discret compatible avec la définition du Chapitre 4 (binaire et respectant la période réfractaire).

Définition 28. [Signal d'entrée discret]

Soit N un neurone. Un signal d'entrée discret pour N est une application $I : \text{Sy}(N) \times \mathbb{R} \rightarrow \mathbb{N}$ tel que pour tout $T \in \mathbb{R}^+$, l'ensemble $\{(s, t) \in \text{Sy}(N) \times [-T, T] \mid I(s, t) \neq 0\}$ est fini. On appelle $\text{Car}(I) = \{(s, t) \in \text{Sy}(N) \times \mathbb{R} \mid I(s, t) \neq 0\}$, le support de I .

On note S_N le monoïde commutatif des signaux d'entrée discrets où l'opération $+$: $S_N \times S_N \rightarrow S_N$ et l'élément neutre 0 sont définis, de manière naturelle, synapse par synapse.

Un signal d'entrée discret I est dit plausible si $I(\text{Sy}(N) \times \mathbb{R}) \subseteq \{0, 1\}$ et si, pour toute synapse $s \in \text{Sy}(N)$, il existe $r_s \in \mathbb{R}^{*+}$ tel que pour tout $t \in \mathbb{R}$, si $I(s, t) = 1$ alors pour tout $t' \in]t, t + r_s[$, $I(s, t') = 0$.

Enfin, un spike sur une synapse s à un temps u est un signal d'entrée discret I_s^u tel que $I_s^u(s, u) = 1$ et $I_s^u(s', t) = 0$ pour tout $(s', t) \neq (s, u)$.

L'ensemble des signaux d'entrée discrets possède une structure de *monoïde*, c'est-à-dire que l'on peut sommer des signaux, que la somme est associative (et ici commutative) et qu'ils possèdent un élément neutre (lorsque le signal vaut 0 partout). Cette structure simplifie des

CHAPITRE 6. UNE VISION PUREMENT ALGÈBRIQUE

preuves, car, pour tout signal d'entrée discret I , on a $I = \sum_{(s,u) \in \text{Car}(I)} I_s^u$. Cette somme est infinie donc on ne peut pas dire que les spikes génèrent le monoïde S_N (par définition, on dit que des éléments g_1, \dots, g_n sont générateurs du monoïde si et seulement si, pour tout élément x du monoïde, x est une somme finie de g_i). Cependant, elle est finie sur tout intervalle de temps borné.

Remarque: Les signaux d'entrée discrets plausibles ne forment pas un sous-monoïde de S_N car la somme de deux signaux d'entrée discrets plausibles ne donne pas forcément un signal d'entrée discret plausible.

Nous définissons la catégorie \mathcal{S} des *signaux d'entrée discrets* (Définition 29 ci-après) dont les objets sont des paires qui associent un neurone et le monoïde des signaux d'entrée discrets pour ce neurone. Les morphismes de \mathcal{S} permettent de comparer les signaux d'entrée discrets pour des neurones entre lesquels il existe un morphisme. S'il existe un morphisme d'un neurone vers un autre alors il partagent forcément les mêmes synapses (du point de vue des symboles) et on peut donc comparer les signaux, synapse par synapse.

Nous définissons également un « morphisme de catégories » ou *foncteur* $F_{\mathcal{S}}$ (Définition 29, illustrée par la Figure 6.8) qui permet de lier la catégorie \mathcal{S} avec la catégorie \mathcal{N} . Le foncteur $F_{\mathcal{S}}$ associe à tout neurone N une paire de la catégorie \mathcal{S} (où le neurone N est préservé), et associe à tout morphisme de neurones $\mu : N \rightarrow N'$ une paire constituée de μ et d'un signal d'entrée continu pour N' .

Définition 29. [Catégorie des signaux d'entrée discrets]

Si $\mu : N \rightarrow N'$ est un morphisme de neurones alors $Sy(N)$ et $Sy(N')$ partagent les mêmes symboles et par conséquent $S_N = S_{N'}$. On note \mathcal{S} la catégorie des signaux d'entrée discrets dont les objets sont les paires (N, S_N) avec $N \in \mathcal{N}$ et dont les morphismes sont les paires (μ, κ) telles que $\mu : N \rightarrow N'$ est un morphisme de neurones et $\kappa : S_N \rightarrow S_{N'}$ est un morphisme de monoïdes.

\mathcal{N} est canoniquement incluse dans \mathcal{S} via le foncteur $F_{\mathcal{S}} : \mathcal{N} \rightarrow \mathcal{S}$ défini par $F_{\mathcal{S}}(N) = (N, S_N)$ pour tout neurone $N \in \mathcal{N}$ et $F_{\mathcal{S}}(\mu) = (\mu, Id_{S_N})$ pour tout morphisme de neurones μ .

$$\begin{array}{ccc}
 N & \xrightarrow{\mu} & N' \\
 \downarrow F_{\mathcal{S}} & & \downarrow F_{\mathcal{S}} \\
 (N, S_N) & \xrightarrow{F_{\mathcal{S}}(\mu)} & (N', S_{N'})
 \end{array}$$

FIGURE 6.8 – Foncteur $F_{\mathcal{S}}$. On représente en rouge ce qui se rapporte à la catégorie des neurones et en gris ce qui se rapporte à la catégorie des monoïdes.

S'il existe un morphisme de neurones $\mu : N \rightarrow N'$, les morphismes de \mathcal{S} permettent de passer d'un signal d'entrée discret S_N pour N à un signal d'entrée discret $S_{N'}$ pour N' . Puisque μ préserve les synapses, le morphisme le plus simple résulte de l'identité sur S_N . Il existe également d'autres morphismes, moins triviaux, dont on détaille certains ci-après.

Morphisme par transformation homothétique

La *transformation homothétique* (μ, κ) où $\kappa : S_N \rightarrow S_{N'}$ multiplie chaque signal d'entrée discret élémentaire par un entier n . La valeur de n peut éventuellement varier d'une synapse à l'autre, de sorte que pour toute synapse s et pour tout t , $\kappa(I)(s, t) = n_s \times I(s, t)$.

Morphisme par décalage de synapses

Le *décalage de synapses* (μ, κ) attribue chaque signal d'entrée discret élémentaire à une synapse de N' qui n'a pas forcément le même symbole que la synapse à laquelle il est attribué dans N . On a ainsi, pour toute synapse s et pour tout t , $\kappa(I)(s, t) = I(f(s), t)$ où $f : Sy(N) \rightarrow Sy(N')$ est une application qui détermine à quelle synapse de N' sera attribué le signal d'entrée discret élémentaire d'une synapse s de N .

Morphisme par shift temporel

Le *shift temporel* (μ, κ) consiste à décaler le signal d'entrée discret dans le temps. On a ainsi, pour toute synapse s et pour tout t , $\kappa(I)(s, t) = I(s, t + \Delta t)$ tel que, si $\Delta t > 0$, le signal d'entrée discret est décalé vers le futur. Δt est un nombre réel qui peut éventuellement varier d'une synapse à l'autre.

Remarque: Lorsque $\mu = Id_N$ et Δt est le même pour toutes les synapses de N , on note Δt -Shift $_N$ le morphisme de \mathcal{S} correspondant.

Nous définissons alors un « morphisme de foncteurs » ou *transformation naturelle* Δt -Shift, qui associe à un signal d'entrée discret pour un neurone, un autre signal d'entrée discret pour ce même neurone (Proposition 2 ci-après, illustrée par la Figure 6.9). Lorsque le signal d'entrée discret cible est obtenu par shift temporel du signal d'entrée discret source, la transformation naturelle permet de définir la dynamique du signal d'entrée discret d'un neurone.

Proposition 2. [Transformation naturelle Δt -Shift]

Pour tout $\Delta t \in \mathbb{R}$, la famille Δt -Shift = $\{\Delta t$ -Shift $_N\}_{N \in \mathcal{N}}$ est une transformation naturelle de $F_{\mathcal{S}}$ vers lui-même.

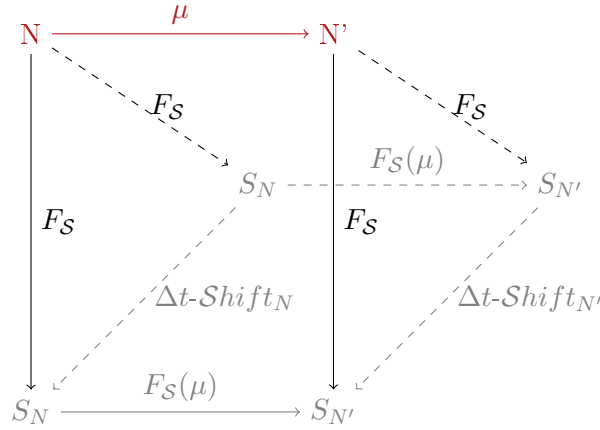


FIGURE 6.9 – Transformation naturelle Δt -Shift. Pour des raisons de lisibilité, on note S_N la paire (N, S_N) . On représente en rouge ce qui se rapporte à la catégorie des neurones et en gris ce qui se rapporte à la catégorie des monoïdes.

Preuve: Il suffit de prouver que le diagramme de la Figure 6.9 est commutatif. Ceci est trivial puisque Δt -Shift $_N$ et Δt -Shift $_{N'}$ sont les mêmes fonctions sur les mêmes ensembles de synapses (car $Sy(N') = Sy(N)$). \square

6.2.2 Signaux d'entrée continus

La fonction des synapses est de convertir un signal d'entrée discret en signaux continus. Ces signaux continus sont appelés « traces » du signal sur les synapses dans le Chapitre 4 (Définition 12). Ici, nous parlerons simplement de *signal d'entrée continu* car nous ne raisonnerons plus sur le signal discret, mais directement sur le signal continu (dédit du signal discret) comme entrée du neurone.

Tout d'abord, rappelons que le produit cartésien de plusieurs corps n'est pas un corps, mais simplement un anneau et qu'un module peut être assimilé à un « espace vectoriel sur un anneau ».

Nous définissons la catégorie Ω des signaux d'entrée continus, de manière similaire à ce que l'on a fait pour les signaux d'entrée discrets. La définition suivante présente les signaux d'entrée continus pour un neurone, comme un *module*. Un morphisme de modules permet alors de passer d'un signal d'entrée continu pour un neurone N à un signal d'entrée continu pour un neurone N' (tel qu'il existe un morphisme de N vers N'). Puisque les fonctions synaptiques des synapses de N' sont q_μ^s fois plus importantes que celles de N (Définition 27), le signal d'entrée continu pour N' sera q_μ^s fois plus important que celui de N .

Nous définissons ci-après un foncteur F_Ω qui permet de lier la catégorie \mathcal{N} avec la catégorie Ω des signaux d'entrée continus. Le foncteur F_Ω associe à tout neurone N une paire de la catégorie Ω (où le neurone N est préservé), et associe à tout morphisme de neurones $\mu : N \rightarrow N'$ une paire constituée de μ et, dans le cas le plus simple, de l'identité sur le

signal d'entrée discret pour N (puisque μ préserve les synapses).

Définition 30. [Catégorie des signaux d'entrée continus]

Soit un neurone N . Un signal d'entrée continu pour N est une application $C : Sy(N) \times \mathbb{R} \rightarrow \mathbb{R}$ tel que, pour toute synapse $s \in Sy(N)$, la fonction $C_s(t) = C(s, t)$ est continue en t .

Nous introduisons la notation préliminaire Λ_N qui correspond à l'anneau défini par le produit de corps $\mathbb{R}^{Sy(N)}$. Un scalaire $\lambda \in \Lambda_N$ est donc une collection de nombres réels $\lambda = \{\lambda_s\}_{s \in Sy(N)}$ indexée par $Sy(N)$ et on note $\lambda.C$ le signal d'entrée continu $C' : Sy(N) \times \mathbb{R} \rightarrow \mathbb{R}$ défini par $C'(s, t) = \lambda_s \times C(s, t)$.

On note Ω_N le module sur l'anneau Λ_N des signaux d'entrée continus pour un neurone donné, où l'opération $+$: $\Omega_N \times \Omega_N \rightarrow \Omega_N$ est définie synapse par synapse de manière naturelle, et la multiplication scalaire est définie comme ci-dessus.

On note Ω la catégorie des signaux d'entrée continus dont les objets sont les paires (N, Ω_N) avec $N \in \mathcal{N}$, et les morphismes sont les paires (μ, κ) telles que $\mu : N \rightarrow N'$ est un morphisme de neurone et $\kappa : \Omega_N \rightarrow \Omega_{N'}$ est un morphisme de module.

De plus, rappelons que si $\mu : N \rightarrow N'$ est un morphisme de neurone alors $Sy(N)$ et $Sy(N')$ partagent le même ensemble de symboles et par conséquent $\Omega_N = \Omega_{N'}$.

Si on considère le scalaire $\lambda_\mu = \{q_\mu^s\}_{s \in Sy(N)}$ (où les q_μ^s sont les quotients de la Définition 27), ce scalaire définit un morphisme de module $Q_\mu : \Omega_N \rightarrow \Omega_{N'}$ avec $Q_\mu(C) = \lambda_\mu.C$. Enfin, on note $F_\Omega : \mathcal{N} \rightarrow \Omega$ le foncteur défini par $F_\Omega(N) = (N, \Omega_N)$ et $F_\Omega(\mu) = (\mu, Q_\mu)$ pour tout neurone $N \in \mathcal{N}$ et pour tout morphisme de neurone $\mu : N \rightarrow N'$.

Les catégories \mathcal{S} et Ω sont liées par l'action des synapses qui convertissent le signal d'entrée discret reçu en signaux continus, qui servent d'entrée aux arbres dendritiques. Ce phénomène peut être modélisé par une transformation naturelle de $F_{\mathcal{S}}$ vers F_Ω (Proposition 3 ci-après, illustrée par la Figure 6.10).

Proposition 3. [Transformation naturelle Convert]

Pour tout neurone N , le module Ω_N est, en particulier, un monoïde commutatif pour l'opération $+$. On définit alors le morphisme de monoïde $Convert_N : S_N \rightarrow \Omega_N$ tel que, pour tout spike I_s^u , $C = Convert_N(I_s^u)$ est le signal d'entrée continu défini par $C(s', t) = 0$ si $s' \neq s$ et $C(s, t) = \sigma_s(t - u)$ pour tout $t \in \mathbb{R}$. La famille $Convert = \{Convert_N\}_{N \in \mathcal{N}}$ est une transformation naturelle de $F_{\mathcal{S}}$ vers F_Ω .

Preuve: Tout d'abord, la définition de $Convert_N$ à partir de l'image de tous les spikes est correcte, car, bien que les spikes ne génèrent pas S_N , la condition de « sparcité » sur les signaux d'entrée discrets nous permet de définir $Convert_N(I)(s, t)$ pour tous les signaux d'entrée discrets I et pour toutes les valeurs finies de t et donc, pour tout $t \in \mathbb{R}$, comme remarqué au début de cette section.

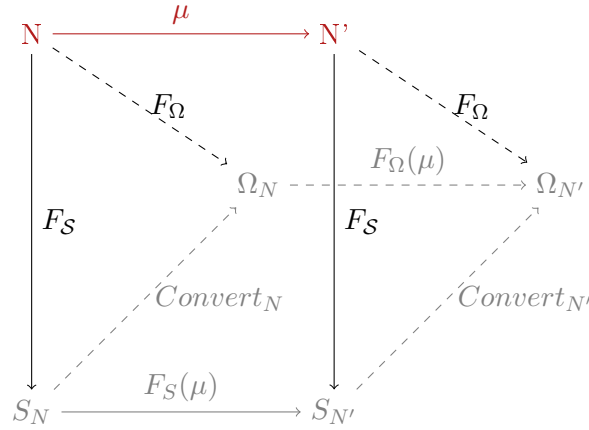


FIGURE 6.10 – Transformation naturelle *Convert*. Pour des raisons de lisibilité, on note S_N la paire (N, S_N) et on note Ω_N la paire (N, Ω_N) . On représente en rouge ce qui se rapporte à la catégorie des neurones et en gris ce qui se rapporte à la catégorie des monoïdes.

Dès lors, pour définir une transformation naturelle à partir de la famille *Convert*, il suffit de prouver la commutativité du diagramme de la Figure 6.10 (c'est-à-dire qu'il suffit de prouver que le résultat de l'application de F_S suivi de l'application de $Convert_{N'}$ est effectivement le même que celui de l'application de $Convert_N$ suivie de l'application de F_Ω). Ceci découle directement des définitions de F_S et F_Ω . Par définition, pour toute synapse s , σ_s est q_μ^s fois plus importante dans N' que dans N , donc le signal continu obtenu par $Convert_{N'}$ est q_μ^s fois plus important que celui obtenu par $Convert_N$. Par ailleurs, le foncteur F_Ω multiplie le signal continu par q_μ^s alors que F_S laisse le signal discret inchangé. Les deux compositions aboutissent donc au même signal d'entrée continu. \square

Puisque *Convert* est une transformation naturelle, étant donné un morphisme $\mu : N \rightarrow N'$, passer d'un signal d'entrée discret pour N à un signal d'entrée discret pour N' (via F_S) puis obtenir un signal d'entrée continu pour N' à partir de ce signal (via $Convert_{N'}$), est équivalent à obtenir un signal d'entrée continu pour N à partir d'un signal d'entrée discret pour N (via $Convert_N$) puis passer à un signal d'entrée continu pour N' à partir de ce signal d'entrée continu pour N (via F_Ω).

6.3 La catégorie des états dendritiques

Pour rappel, l'état d'un neurone est la valeur de la charge en tous ses points (Définition 8, Chapitre 4). De plus, l'état dendritique d'un neurone est la valeur de la charge aux synapses et en tout point des compartiments, permettant de s'affranchir de la charge au soma (Définition 9, Chapitre 4). Notre définition des compartiments abstrait l'espace en temps, en caractérisant chaque compartiment c par un délai Δ_c couplé à un facteur d'atténuation α_c (Définitions 4, Chapitre 4). De plus, la valeur de la charge en entrée d'un compartiment est la somme des charges sortant de ses prédécesseurs. Dans ce chapitre, on redéfinit l'état dendritique (Définition 31 ci-après) pour exploiter de manière cohérente la nouvelle

6.3. LA CATÉGORIE DES ÉTATS DENDRITIQUES

définition des synapses. Effectivement, toute synapse s est dorénavant munie d'un facteur d'atténuation ($\alpha_s = 1$ par convention) et d'un délai ($\Delta_s = 0$ par convention) (Notation 3).

Définition 31. [Catégorie des états dendritiques d'un neurone]

Étant donné un neurone N , un état dendritique de N est une famille $\partial = \{\eta_x\}_{x \in Sy(N) \cup Co(N)}$ indexée par $Sy(N) \cup Co(N)$ telle que :

- pour tout $x \in Sy(N) \cup Co(N)$, $\eta_x : [0, \Delta_x] \rightarrow \mathbb{R}$ est une fonction continue ;
- pour tout $c \in Co(N)$, $\eta_c(\Delta_c) = \left(\sum_{x \in Pred_N(c)} \eta_x(0) \right) \times \alpha_c$.

Par extension, on note souvent $\eta_{\nabla}(0) = \sum_{x \in Pred_N(\nabla)} \eta_x(0)$, de sorte que ∂ puisse également être vu comme une famille indexée par $Sy(N) \cup Co(N) \cup \{\nabla\}$.

On note D_N le groupe abélien des états dendritiques de N où l'opération $+$: $D_N \times D_N \rightarrow D_N$ est définie nœud par nœud, de manière naturelle. On note \mathcal{D} la catégorie des états dendritiques dont les objets sont les paires (N, D_N) où $N \in \mathcal{N}$, et les morphismes sont les paires (μ, κ) telles que $\mu : N \rightarrow N'$ est un morphisme de neurones et $\kappa : D_N \rightarrow D_{N'}$ est un morphisme de groupe.

Similairement au travail présenté dans le Chapitre 5, nous souhaitons construire l'état dendritique d'un neurone N' à partir de l'état dendritique d'un neurone N tel qu'il existe un morphisme de N vers N' . La vision algébrique que nous adoptons ici permet de passer simplement par un foncteur. Dans le but de définir un foncteur $F_{\mathcal{D}} : \mathcal{N} \rightarrow \mathcal{D}$ compatible avec $F_{\mathcal{S}}$ et F_{Ω} (Définition 32 plus loin), nous définissons ci-après quelques notations techniques et propriétés. Dans un premier temps, nous définissons les *prédécesseurs par morphismes* d'un nœud et établissons leurs propriétés (Proposition 4 ci-après). Étant donné un morphisme $\mu : N \rightarrow N'$, les prédécesseurs par morphismes d'un nœud y de N' sont les nœuds de N dont l'état va contribuer à l'état de y .

Remarque: Lorsqu'un compartiment y de N' n'existe pas dans N , les compartiments de N qui contribuent à l'état de y se retrouvent forcément à une profondeur plus importante que y dans l'arbre sous-jacent de N' (Définition 27). C'est pour cette raison qu'on utilise le terme prédécesseur par morphisme (Figure 6.11).

Proposition 4. [Propriétés des prédécesseurs par morphismes]

Soient $\mu : N \rightarrow N'$ un morphisme de neurones et y un nœud de N' . On note $Pred_{\mu}(y)$ l'ensemble des nœuds $x \in Nodes(N)$ tels que le chemin $]\mu(x)_{\xrightarrow{N'} y}]$ existe et ne contient aucune image par μ (il n'y a pas de z tel que $\mu(z) \in]\mu(x)_{\xrightarrow{N'} y}]$).

De plus, si z est un nœud de N , on note $SyPre_N(z) = \{s \in Sy(N) \mid]s_{\xrightarrow{N} z}] \text{ existe}\}$.

Les ensembles $Pred_{\mu}(y)$ et $SyPre_N(z)$ ont les propriétés suivantes :

1. ils ne sont pas vides ;
2. s'il existe $x \in Nodes(N)$ tel que $y = \mu(x)$, alors $Pred_{\mu}(y) = \{x\}$;
3. pour tout $x \in Pred_{\mu}(y)$, $\Delta_y + \Sigma_{N'}^{y \rightarrow \nabla} \leq \Delta_x + \Sigma_N^{x \rightarrow \nabla}$;

4. pour tout $x \in \text{Pred}_\mu(y)$, $\Sigma_N^{x \rightarrow \nabla} \leq \Sigma_{N'}^{y \rightarrow \nabla}$ et donc $\Delta_x \geq \Delta_{\mu(x)} + \Sigma_{N'}^{x \rightarrow y}$;
5. si $z \in \text{Sy}(N)$ alors $\text{SyPre}_N(z) = \{z\}$, sinon pour toute $s \in \text{SyPre}_N(z)$, $\Sigma_N^{s \rightarrow \nabla} - \Sigma_N^{z \rightarrow \nabla} \geq \Delta_z > 0$;
6. $\bigcup_{x \in \text{Pred}_\mu(y)} \text{SyPre}_N(x) = \text{SyPre}_{N'}(y)$.

Preuve: Ces propriétés résultent principalement de la Définition 27.

1. $\text{SyPre}_N(x)$ n'est pas vide, car, par définition, toutes les feuilles de N sont des synapses. $\text{Pred}_\mu(y)$ n'est pas vide, car il existe au moins une synapse s telle que $]s_{N'} \rightarrow y]$ existe et donc, si on considère le nœud x de N tel que $\mu(x)$ est la plus proche image de y via μ sur le chemin $]s_{N'} \rightarrow y]$, x existe forcément, car il peut être égal à s s'il n'y a pas d'autre image par μ dans $]s_{N'} \rightarrow y]$ et donc $x \in \text{Pred}_\mu(y)$.
2. Évident (car il n'y a pas d'image par μ entre $y = \mu(x)$ et $\mu(x)$).
3. Cette propriété découle de $\Delta_x + \Sigma_N^{x \rightarrow \nabla} \leq \Delta_{\mu(x)} + \Sigma_{N'}^{\mu(x) \rightarrow \nabla}$ puisque y qui appartient à $]\mu(x)_{N'} \rightarrow \nabla]$.
4. Prenons le successeur u de x dans N et un nœud y tel que $\mu(u)$ appartient à $]y_{N'} \rightarrow \nabla]$ et donc, $\Delta_{\mu(u)} + \Sigma_{N'}^{\mu(u) \rightarrow \nabla} \leq \Sigma_{N'}^{y \rightarrow \nabla}$ car $\Delta_{\mu(u)} + \Sigma_{N'}^{\mu(u) \rightarrow \nabla} = \Delta_u + \Sigma_N^{u \rightarrow \nabla} = \Sigma_N^{x \rightarrow \nabla}$.
5. Évident.
6. Évident, puisque N et N' ont les mêmes synapses (symboles).

□

Remarque: Ne pas confondre $\text{SyPre}_N(z)$ avec $\text{SyPred}_N(z)$ qui est l'ensemble des synapses prédécesseurs directs de z dans N .

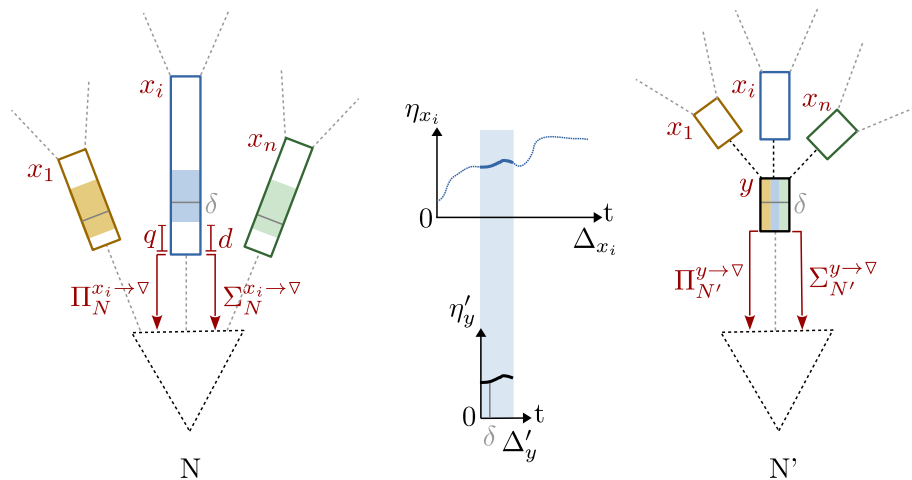


FIGURE 6.11 – Illustration de la Définition 32. Le morphisme de groupe $\bar{\mu} : D_N \rightarrow D_{N'}$ associe à l'état dendritique d'un neurone N (partie gauche de la figure) l'état dendritique d'un neurone N' (partie droite de la figure), étant donné le morphisme de neurones $\mu : N \rightarrow N'$. L'état de tout nœud y de N' est calculé à partir de l'état de ses prédécesseurs par morphisme dans N (ici, l'état des x_i). On note $d = \Sigma_{N'}^{y \rightarrow \nabla} - \Sigma_N^{x \rightarrow \nabla}$ et $q = \frac{\Pi_N^{x \rightarrow \nabla}}{\Pi_{N'}^{y \rightarrow \nabla}}$.

6.3. LA CATÉGORIE DES ÉTATS DENDRITIQUES

Nous pouvons dorénavant définir le morphisme de groupe $\bar{\mu} : D_N \rightarrow D_{N'}$ qui permet de calculer l'état dendritique d'un neurone N' à partir de l'état dendritique d'un neurone N tel qu'il existe un morphisme μ de N vers N' (Définition 32 ci-après, illustrée par la Figure 6.11). L'état de chaque compartiment de N' est calculé à partir de l'état de ses prédécesseurs par morphisme dans N .

Le foncteur $F_{\mathcal{D}} : \mathcal{N} \rightarrow \mathcal{D}$ associe à tout neurone de \mathcal{N} une paire de la catégorie \mathcal{D} (où le neurone N est préservé), et associe à tout morphisme de neurones $\mu : N \rightarrow N'$ une paire constituée de μ et de $\bar{\mu}$.

Définition 32. [Foncteur $F_{\mathcal{D}}$]

Soit $\mu : N \rightarrow N'$ un morphisme de neurones. On définit $\bar{\mu} : D_N \rightarrow D_{N'}$ comme le morphisme de groupe qui associe à chaque état dendritique $\partial = \{\eta_x\}_{x \in Sy(N) \cup Co(N)}$ de N l'état dendritique $\partial' = \{\eta'_y\}_{y \in Sy(N') \cup Co(N')}$ de N' , où pour tout $y \in Sy(N') \cup Co(N')$, $\eta'_y : [0, \Delta_y] \rightarrow \mathbb{R}$ est défini comme suit, pour tout $\delta \in [0, \Delta_y]$:

$$\eta'_y(\delta) = \sum_{x \in Pred_{\mu}(y)} \eta_x(\delta + \Sigma_{N'}^{y \rightarrow \nabla} - \Sigma_N^{x \rightarrow \nabla}) \times \frac{\Pi_N^{x \rightarrow \nabla}}{\Pi_{N'}^{y \rightarrow \nabla}} \times q_{\mu}^{\beta}$$

On définit ensuite $F_{\mathcal{D}} : \mathcal{N} \rightarrow \mathcal{D}$ comme le foncteur tel que $F_{\mathcal{D}}(N) = (N, D_N)$ pour tout neurone $N \in \mathcal{N}$ et $F_{\mathcal{D}}(\mu) = (\mu, \bar{\mu})$ pour tout morphisme de neurones $\mu : N \rightarrow N'$.

Remarque: D'après la Proposition 4, on a $0 \leq \Sigma_{N'}^{y \rightarrow \nabla} - \Sigma_N^{x \rightarrow \nabla} \leq \delta + \Sigma_{N'}^{y \rightarrow \nabla} - \Sigma_N^{x \rightarrow \nabla} \leq \Delta_y + \Sigma_{N'}^{y \rightarrow \nabla} - \Sigma_N^{x \rightarrow \nabla} \leq \Delta_x$, donc η'_y est bien défini. De plus, $\bar{\mu}$ préserve l'opération $+$ sur les états.

Appliquer un signal d'entrée continu aux synapses d'un neurone est un moyen « d'initialiser » l'état dendritique d'un neurone (Figure 6.13). On définit dans ce cadre, le morphisme de groupe $Init_N : \Omega_N \rightarrow D_N$ qui associe à tout signal d'entrée continu pour un neurone N un état dendritique de N . Il en découle que la famille $\{Init_N\}_{N \in \mathcal{N}}$ est une transformation naturelle :

Proposition 5. [Transformation naturelle $Init$]

Pour tout neurone N , le module Ω_N est en particulier un groupe commutatif pour $+$. On définit alors $Init_N : \Omega_N \rightarrow D_N$ comme le morphisme de groupe qui associe à chaque signal d'entrée continu $C : Sy(N) \times \mathbb{R} \rightarrow \mathbb{R}$ l'état dendritique $\partial = \{\eta_x\}_{x \in Sy(N) \cup Co(N)}$ où $\eta_x : [0, \Delta_x] \rightarrow \mathbb{R}$ est défini comme suit. Pour tout $\delta \in [0, \Delta_y]$:

$$\eta_x(\delta) = \sum_{s \in SyPre_N(x)} C(s, \delta + \Sigma_N^{x \rightarrow \nabla} - \Sigma_N^{s \rightarrow \nabla}) \times \frac{\Pi_N^{s \rightarrow \nabla}}{\Pi_N^{x \rightarrow \nabla}}$$

La famille $Init = \{Init_N\}_{N \in \mathcal{N}}$ est une transformation naturelle de F_{Ω} vers $F_{\mathcal{D}}$.

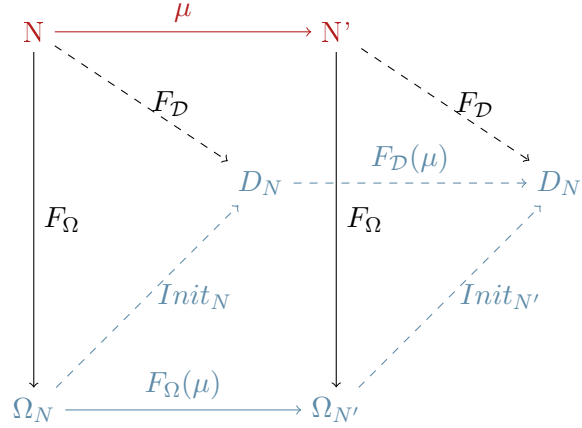


FIGURE 6.12 – Transformation naturelle *Init*. Pour des raisons de lisibilité, on note Ω_N la paire (N, Ω_N) et on note D_N la paire (N, D_N) . On représente en rouge ce qui se rapporte à la catégorie des neurones et en bleu ce qui se rapporte à la catégorie des groupes.

Preuve: Il faut prouver que le diagramme de la Figure 6.12 commute. Il suffit donc de prouver que pour tout nœud $y \in N'$ et pour tout $\delta \in [0, \Delta_y]$, on a :

$$\bar{\mu}(Init_N(C))_y(\delta) = \sum_{s \in SyPre_{N'}(y)} C(s, \delta + \Sigma_{N'}^{y \rightarrow \nabla} - \Sigma_{N'}^{s \rightarrow \nabla}) \times q_\mu^s \times \frac{\Pi_{N'}^{s \rightarrow \nabla}}{\Pi_{N'}^{y \rightarrow \nabla}}$$

Par définition :

$$\bar{\mu}(Init_N(C))_y(\delta) = \sum_{x \in Pred_\mu(y)} \eta_x(s, \delta + \Sigma_{N'}^{y \rightarrow \nabla} - \Sigma_N^{x \rightarrow \nabla}) \times \frac{\Pi_N^{x \rightarrow \nabla}}{\Pi_{N'}^{y \rightarrow \nabla}} \times q_\mu^\beta \text{ où } \eta = Init_N(C).$$

Donc :

$$\eta_x(\delta + \Sigma_{N'}^{y \rightarrow \nabla} - \Sigma_N^{x \rightarrow \nabla}) = \sum_{s \in SyPre_N(x)} C(s, \delta + \Sigma_{N'}^{y \rightarrow \nabla} - \Sigma_N^{s \rightarrow \nabla}) \times q_\mu^s \times \frac{\Pi_N^{s \rightarrow \nabla}}{\Pi_N^{x \rightarrow \nabla}}.$$

Par conséquent :

$$\bar{\mu}(Init_N(C))_y(\delta) = \sum_{x \in Pred_\mu(y)} \sum_{s \in SyPre_N(x)} C(s, \delta + \Sigma_{N'}^{y \rightarrow \nabla} - \Sigma_N^{s \rightarrow \nabla}) \times \frac{\Pi_N^{s \rightarrow \nabla}}{\Pi_N^{x \rightarrow \nabla}} \times \frac{\Pi_N^{x \rightarrow \nabla}}{\Pi_{N'}^{y \rightarrow \nabla}} \times q_\mu^\beta.$$

D'après la Proposition 4 :

$$\bar{\mu}(Init_N(C))_y(\delta) = \sum_{s \in SyPre_{N'}(y)} C(s, \delta + \Sigma_{N'}^{y \rightarrow \nabla} - \Sigma_N^{s \rightarrow \nabla}) \times \frac{\Pi_N^{s \rightarrow \nabla}}{\Pi_{N'}^{y \rightarrow \nabla}} \times q_\mu^\beta.$$

Or d'après la Définition 27, on a $q_\mu^s \times \Pi_{N'}^{s \rightarrow \nabla} = \Pi_N^{s \rightarrow \nabla} \times q_\mu^\beta$, ce qui prouve la proposition. \square

Remarque: La valeur de $(\delta + \Sigma_N^{x \rightarrow \nabla} - \Sigma_N^{s \rightarrow \nabla})$ est toujours négative : seuls les temps passés du signal d'entrée continu servent à l'initialisation de l'état du neurone. Effectivement, l'état du neurone est la conséquence des spikes précédemment reçus aux synapses.

Puisque *Init* est une transformation naturelle, étant donné un morphisme $\mu : N \rightarrow N'$, passer d'un signal d'entrée continu pour N à un signal d'entrée continu pour N' (via F_Ω) puis initialiser l'état dendritique de N' avec ce signal (via $Init_{N'}$), est équivalent à initialiser l'état dendritique de N avec un signal d'entrée continu pour N (via $Init_N$) puis construire l'état dendritique de N' à partir de celui de N (via F_Ω).

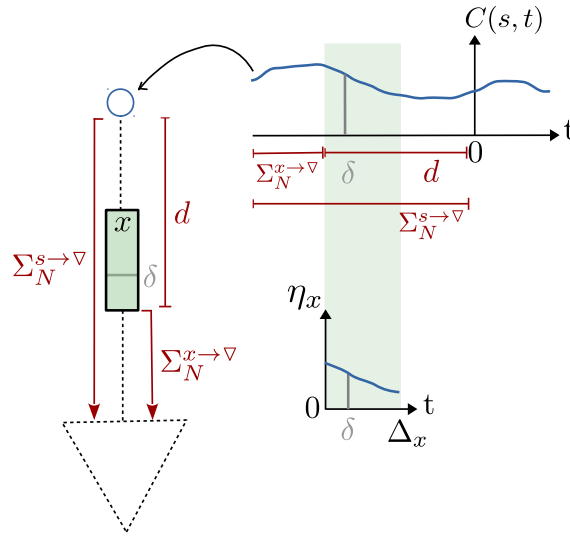


FIGURE 6.13 – *Illustration de la Proposition 5. Un signal d'entrée continu peut permettre d'initialiser l'état dendritique d'un neurone N . L'état ainsi obtenu est alors en continuité avec le signal. On note $d = \Sigma_N^{s \rightarrow \nabla} - \Sigma_N^{x \rightarrow \nabla}$.*

6.4 Préservation de la fonction d'entrée/sortie des neurones par morphisme

Dans ce chapitre, nous avons tiré profit de la théorie des catégories. C'est un langage qui permet de formaliser les relations entre les différents « objets » de notre cadre de modélisation. Nous avons ainsi défini des morphismes entre les neurones, qui lient les différentes structures. Nous avons également défini des morphismes entre les signaux d'entrée discrets. En particulier, il est possible de définir une dynamique, en passant d'un signal d'entrée discret à un autre qui représente son décalage dans le temps. Par la définition du foncteur F_S , nous avons établi une relation entre les neurones et les signaux d'entrée discrets qui peuvent être reçus à leurs synapses. Similairement, le foncteur F_Ω relie les neurones aux signaux d'entrée continus qui peuvent être reçus à leurs synapses. Enfin, les transformations naturelles nous ont permis de mettre en évidence le diagramme commutatif de la Figure 6.14. En particulier, nous avons prouvé que F_S et Δt -Shift commutent. Ceci signifie que « shifter » le signal d'entrée discret de N puis obtenir le signal d'entrée discret de N' via F_S , est équivalent à obtenir le signal d'entrée de N' puis le « shifter ».

Le diagramme de la Figure 6.14 montre que :

1. « shifter » le signal d'entrée discret de N puis,
2. convertir le signal d'entrée discret de N en un signal continu puis,
3. initialiser l'état dendritique de N avec le signal continu obtenu puis,
4. construire l'état dendritique de N' à partir de celui de N ,

est équivalent à :

1. obtenir le signal d'entrée discret de N' à partir de celui de N puis,
2. « shifter » le signal d'entrée discret de N' puis,
3. convertir le signal d'entrée discret de N' en un signal continu puis,
4. initialiser l'état dendritique de N' avec le signal continu obtenu.

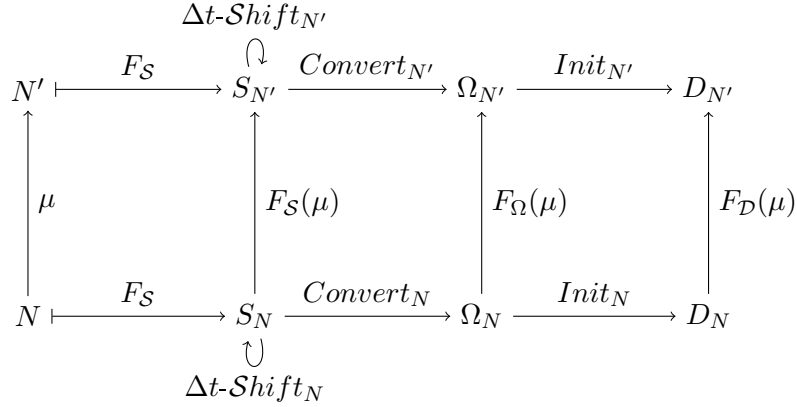


FIGURE 6.14 – Diagramme commutatif.

En d'autres mots, l'état dendritique de N' obtenu par initialisation à partir du signal d'entrée « shifté » de N' est égal à celui obtenu par construction à partir de l'état dendritique de N initialisé par signal d'entrée « shifté » de N . Puisque la dynamique dendritique peut se faire par une succession de shifts, l'état dendritique de N' peut être construit à partir de l'état dendritique de N , à tout temps. On retrouve ici les propriétés de commutativité du Chapitre 5.

Par définition, l'état dendritique de N' construit à partir de celui de N est, en tout point, q_μ^β fois plus important que dans N (Définition 32). Or, également par définition, le seuil de N' est également q_μ^β fois supérieur à celui de N (Définition 27). De plus, N et N' ont le même coefficient de fuite. Par conséquent, N et N' atteindront toujours leur seuil en même temps, produisant la même sortie. On retrouve donc ici le théorème fondamental du Chapitre 5, mais étendu à tout morphisme, et pas seulement si N est minimal.

La vision algébrique permet donc de généraliser les théorèmes du Chapitre 5, tout en facilitant les preuves. Pour rappel, nous avons prouvé que tout neurone a la même fonction d'entrée/sortie que ses formes réduites et normalisées. Dans ce chapitre, nous mettons en évidence l'existence d'un « continuum » de neurones partageant la même fonction d'entrée/sortie, dont les structures sont liées par des morphismes et les états par des foncteurs. Parmi ces neurones, nous avons montré qu'il existe toujours un neurone minimal qui représente canoniquement tout le continuum. Les neurones minimaux correspondent, en fait, à la forme normalisée des neurones. Ceci confirme que la forme normalisée représente canoniquement toute une classe d'équivalence de structures.

Dans cette section, nous n'allons cependant pas jusqu'à donner une vision catégorique des états du soma et des signaux de sortie. Effectivement, l'équation différentielle qui décrit

6.4. PRÉSERVATION DE LA FONCTION D'ENTRÉE/SORTIE DES NEURONES PAR MORPHISME

la dynamique du soma n'a pas de propriétés algébriques intéressantes. Une vision catégorique du soma nécessiterait de se placer dans une catégorie de fonctions continues ce qui n'apporterait rien quant à la facilité de preuves.

Sur le fond, tous les résultats présentés ici découlent de la définition des morphismes de neurones (Définition 27) qui a été choisie avec soin.

Troisième partie

IMPLÉMENTATION DISCRÈTE DU CADRE DE MODÉLISATION ET EXTENSION VERS LA PLASTICITÉ

Dans cette partie, nous traitons de la mise en pratique du cadre de modélisation que nous avons introduit dans la partie précédente.

Nous choisissons d'implémenter notre modèle de neurone en Lustre qui est un langage synchrone basé sur les flots, particulièrement adapté à notre système. Cette implémentation nécessite une discrétisation préalable de notre modèle hybride.

Enfin, dans un chapitre prospectif, nous étendons notre modèle en introduisant de la plasticité synaptique, un phénomène crucial dans la formation des assemblées de neurones.



CHAPITRE 7

DISCRÉTISATION TEMPORELLE DU MODÈLE

Afin d'implémenter notre modèle de neurone en Lustre (Chapitre 2, Section 2.4), notamment en vue d'y appliquer du model checking, nous devons discrétiser le temps. Le temps est alors « découpé » en petits intervalles (pas de temps). Nous verrons dans le chapitre suivant que la succession des pas de temps constitue la suite des instants logiques abstraits caractéristiques des langages synchrones.

Pour faciliter la discrétisation et les preuves par model checking subséquentes, on considère dans ce chapitre uniquement les cas linéaires par morceaux des fonctions synaptiques et des fonctions de bornes (Figures 4.6 et 4.3). La linéarité de ces équations est, par définition, préservée au cours de la propagation des signaux dans les dendrites (Chapitre 4, Section 4.2, Définition 15).

Lorsque l'on souhaite préserver l'essence du fonctionnement biologique, l'intégration au soma induit inévitablement une équation différentielle. Malheureusement, cette équation différentielle, sans une approximation telle que celle obtenue par discrétisation du temps, compromettrait l'approche par model checking. La discrétisation est, de fait, essentielle pour la suite.

L'objet de notre étude est la fonction d'entrée/sortie des neurones. Puisque nous avons prouvé que la réduction des neurones préserve cette fonction (Chapitre 5), nous choisissons de n'implémenter des neurones que sous forme réduite. Comme expliqué dans le chapitre précédent, la représentation sous forme réduite (et éventuellement normalisée) simplifie largement l'implémentation et le calcul de la dynamique.

7.1 Choix du pas de temps

Une étape clé de la discrétisation est le choix du pas de temps. De manière générale (Chapitre 2, Section 2.1), plus le pas de temps est petit et plus l'approximation s'avère précise. En revanche, le choix d'un petit pas s'accompagne d'une augmentation du coût calculatoire. Le choix du pas est donc un compromis qui dépend du phénomène étudié.

Dans notre contexte, des événements sont généralement définis comme étant synchrones lorsqu'ils se produisent ensemble dans une fenêtre temporelle de 1 à 5 millisecondes [Grün et al., 1999] [Izhikevich, 2006] [Brette, 2012]. Le pas de temps que nous devons choisir ne doit donc pas dépasser cet ordre de grandeur afin de pouvoir discriminer convenablement des spikes synchrones de spikes asynchrones. De plus, afin d'observer les processus de sommation, ce pas doit être bien inférieur aux paramètres synaptiques $\hat{\tau}$ et $\tilde{\tau}$.

Pour les simulations de modèles très détaillés, qui s'intéressent à des phénomènes aussi précis que la forme des potentiels d'actions, un pas de 0.01 milliseconde semble approprié [Börger and Nectow, 2013] [Ratnadurai Giridharan et al., 2014] [Zadeh and Kambhampati, 2017]. Ici, on considère des spikes discrets et cette approximation montre bien qu'un pas de temps si petit ne nous est pas nécessaire. Dans notre contexte, un pas de temps de 0.1 milliseconde semble approprié et en accord avec les simulations de modèles de type Integrate-and-Fire que l'on peut trouver dans la littérature [Salinas and Sejnowski, 2002] [Brette and Goodman, 2008] [Nordlie et al., 2010] [Orhan, 2012] [Van Rossum, 2014] [goldman lab, 2018].

Il s'agit là d'un ordre de grandeur et la valeur exacte du pas de temps (que l'on note δ) peut naturellement être modifiée au besoin : le processus de discrétisation reste alors le même. La discrétisation consiste finalement à exprimer tous les paramètres temporels du modèle en multiples de δ .

7.2 Discrétisation aux synapses

Chaque synapse est caractérisée par une fonction synaptique (Chapitre 4, Section 4.1, Définition 2). Étant donnée une synapse s , cette fonction, lorsqu'elle est linéaire, est définie par deux paramètres temporels et son poids : le temps durant lequel la valeur absolue de la fonction est croissante ($\hat{\tau}_s$) jusqu'à atteindre sa valeur maximale $|w_s|$, et le temps durant lequel elle est décroissante ($\tilde{\tau}_s$).

Pour discrétiser le temps au niveau des synapses, il suffit d'exprimer ces paramètres temporels, $\hat{\tau}_s$ et $\tilde{\tau}_s$, en multiples de δ . Lors de la réception d'un spike à la synapse s , la valeur de la charge locale évolue alors de manière discontinue par sauts de $\frac{w_s}{\tilde{\tau}}$ à chaque pas de temps δ jusqu'à atteindre la valeur w_s , puis par sauts de $\frac{w_s}{\tilde{\tau}}$ jusqu'au retour à 0 (Figure 7.1).

Le paramètre $\hat{\tau}_s$ (auss appelé *time-to-peak*) varie entre une et plusieurs dizaines de mil-

liseconds et le paramètre $\tilde{\tau}_s$ est généralement plus grand [Williams and Stuart, 2002] [Magee and Cook, 2000]. Ceci fait de 0.1 milliseconde, un pas de temps approprié puisque le nombre de points considérés est suffisant pour modéliser raisonnablement la trace discrétisée du signal (Figure 7.1).

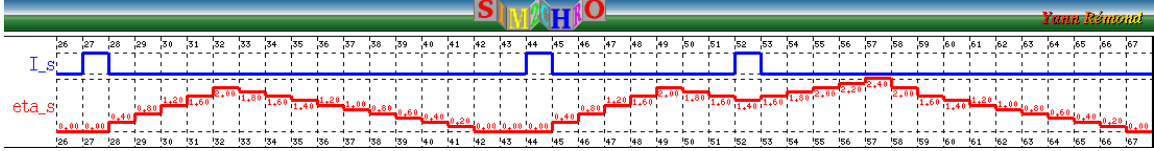


FIGURE 7.1 – Simulation de l'état η_s d'une synapse excitatrice s en réponse à un signal d'entrée I_s . La synapse s possède les paramètres arbitraires $\hat{\tau}_s = 5$, $\tilde{\tau}_s = 10$ et $w_s = 2$. On observe le phénomène de sommation temporelle lorsque deux spikes reçus sont suffisamment proches dans le temps. La synapse s sous forme discrétisée est encodée en Lustre et simulée par l'outil Luciole.

7.3 Discrétisation aux compartiments

Les compartiments sont caractérisés par un temps de traversée Δ et un facteur d'atténuation α (Chapitre 4, Section 4.1, Définition 4). Nous devons alors discrétiser le paramètre temporel Δ en un multiple de δ . Tout compartiment dont le temps de traversée est inférieur au pas de temps δ sera donc considéré nul. Ceci peut amener à regrouper des synapses, que l'on peut assimiler aux clusters observés expérimentalement [Stuart et al., 2016] [Gökçe et al., 2016].

La discrétisation au niveau des compartiments consiste à découper chaque compartiment $c = (\alpha_c, \Delta_c)$ en $n = \frac{\Delta_c}{\delta}$ sous-compartiments de temps de traversée δ . Un facteur d'atténuation doit être défini pour chaque sous-compartiment de manière à ce que l'atténuation globale α_c soit préservée à l'issue de c (Figure 7.2).



FIGURE 7.2 – Discrétisation aux compartiments. Le compartiment $c = (\alpha_c, \Delta_c)$ est découpé en sous-compartiments de manière à préserver le temps de traversée et l'atténuation globale à l'issue de c .

Si le neurone considéré est sous forme normalisée, α_c est toujours égal à 1 par définition, ce qui signifie que le signal ne subit pas d'atténuation au cours de la traversée du compartiment c . Il suffit donc de donner la valeur 1 au facteur d'atténuation de chaque sous-compartiment pour préserver ce comportement. En revanche, si le neurone est réduit, mais qu'il n'est pas normalisé, le facteur d'atténuation de chaque sous-compartiment dépend des paramètres du compartiment d'origine. On peut supposer chaque compartiment homogène sans aucune perte de généralité, de sorte que l'atténuation se fait de la même manière dans tous ses sous-

compartiments. Étant donné un compartiment $c = (\alpha_c, \Delta_c)$, le processus de discrétisation produit donc n sous-compartiments $subC = (\alpha_{sub}, \Delta_{sub})$ où $\Delta_{sub} = \delta$ et $\alpha_{sub} = \sqrt[n]{\alpha_c}$ (Figure 7.2).

Avec cette représentation discrétisée des compartiments, la dynamique peut être calculée par glissement de tous les sous-compartiments à chaque pas de temps, puisqu'ils ont la même durée δ . Cette solution est alors exacte.

7.4 Discrétisation au soma

Le potentiel au soma évolue selon une équation différentielle du premier ordre (Chapitre 4, Section 4.2, Définition 16). Fondamentalement, il s'agit de remplacer l'équation différentielle par une somme à calculer à chaque pas de temps δ . La méthode d'*Euler explicite* (Chapitre 2, Section 2.1) est la technique la plus simple et la plus intuitive pour atteindre cet objectif.

D'après le lemme 3 (Chapitre 4, Section 4.2), le potentiel de soma P est gouverné par l'équation suivante : $P'(t) = \frac{dP}{dt} = F(t) - \gamma.P(t)$. En appliquant la méthode d'Euler avec un pas de temps δ , on obtient l'équation 7.1.

$$P(t + \delta) = P(t) + \delta.P'(t) = P(t) + \delta(F(t) - \gamma.P(t)) \quad (7.1)$$

Cette solution numérique permet alors de calculer le potentiel au soma à chaque pas de temps $(t + \delta)$, à partir de sa valeur au pas précédent (t) .

La méthode d'Euler est loin d'être la plus précise, mais il s'agit là d'une simple *preuve de concept* de la discrétisation de notre cadre de modélisation. On pourrait également obtenir une solution numérique de l'équation différentielle par d'autres méthodes plus sophistiquées comme la méthode *RK4* par exemple (Chapitre 2, Section 2.1).

Nous décrivons ici l'évolution sous-liminaire du potentiel qui est continue. Or, la dynamique du soma contient des discontinuités aux temps des spikes, lorsque le potentiel atteint son seuil. Ces temps seront également approximés (retardés) et pourraient entraîner une accumulation d'erreur. Néanmoins, le pas de temps que nous choisissons est relativement petit ce qui la limite grandement.

Remarque: La discrétisation de la fonction de borne, qui détermine la valeur du seuil au cours du temps, se fait de manière similaire à la discrétisation de la fonction synaptique (Chapitre 8, Section 8.1).

CHAPITRE 8

PROTOTYPAGE AVEC LUSTRE

Sur le conseil, et avec l'aide, de *Daniel Gaffé*, du *Laboratoire d'Électronique, Antennes et Télécommunication (LEAT, Sophia Antipolis)*, nous avons choisi d'implémenter notre modèle en Lustre (Chapitre 2, Section 2.4). Lustre est un langage de programmation très adapté à l'encodage de notre système. Effectivement, le neurone est système réactif synchrone : il interagit continuellement avec son environnement et y répond immédiatement. De plus, Lustre est un langage de flots, ce qui permet de représenter facilement les signaux d'entrée des neurones ainsi que les charges qui s'y déplacent. Enfin, Lustre bénéficie d'une définition formelle. Ce langage déclaratif possède donc une sémantique mathématique précise, levant toute ambiguïté, et cette sémantique permet l'application du model checking.

8.1 Encodage du modèle

Nous implémentons notre modèle sous sa forme discrétisée, décrite dans le chapitre précédent (Chapitre 7). Chaque composant du neurone (synapse, compartiment, soma) est encodé sous forme d'un nœud Lustre.

8.1.1 Encodage d'une synapse

Une synapse s est un nœud qui calcule une sortie réelle `eta_s` à partir d'un flot booléen `I_s` (Figure 8.1). `I_s` représente le signal d'entrée à la synapse s qui vaut `true` aux instants des spikes et `false` à tous les autres instants. La valeur de `eta_s` correspond à l'état de la synapse à chaque instant logique. Elle est déduite de la trace de `I_s` sur s , qui dépend elle-même de la fonction synaptique caractéristique de s (Chapitre 4, Section 4.2, Définition 12). Dans le cadre de l'approximation linéaire que l'on considère ici, elle dépend directement des paramètres w (w_s), `hatTau` ($\hat{\tau}_s$) et `checkTau` ($\check{\tau}_s$) qui la définissent. Ces paramètres

correspondent respectivement à la valeur maximale de la variation de charge déclenchée par un spike, au temps de montée et au temps de descente de la charge.

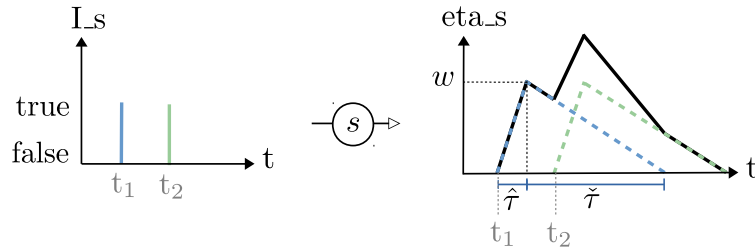


FIGURE 8.1 – Calcul de l'état eta_s d'une synapse s en fonction d'un signal I_s . La trace du signal sur s dépend des paramètres w (w), $\hat{\tau}$ ($\hat{\tau}$) et $\check{\tau}$ ($\check{\tau}$) caractéristiques de s .

De par l'absence de boucles en Lustre, un nœud est requis pour tout couple de paramètres ($\hat{\tau}$, $\check{\tau}$). Prenons l'exemple simple d'un nœud qui encode une synapse excitatrice de paramètres $w = 1$, $\hat{\tau} = 2$ et $\check{\tau} = 4$ (Nœud 1). Notons que ces paramètres sont choisis ici pour des raisons pédagogiques, sans réelle plausibilité biologique.

Nœud 1: Synapse excitatrice de paramètres $w_s = 1$, $\hat{\tau}_s = 2$ et $\check{\tau}_s = 4$.

```
node synapse (I_s : bool) returns (eta_s : real);
  var w, hatTau, checkTau : real;
      o1, o2, o3, o4, o5 : bool;
  let
    w = 1.0;
    hatTau = 2.0;
    checkTau = 4.0;
    o1 = false -> pre(I_s);
    o2 = false -> pre(o1);
    o3 = false -> pre(o2);
    o4 = false -> pre(o3);
    o5 = false -> pre(o4);
    eta_s = 0.0 -> (if o1 then w/hatTau else 0.0)
                  + (if o2 then 2.0*w/hatTau else 0.0)
                  + (if o3 then w-w/checkTau else 0.0)
                  + (if o4 then w-2.0*w/checkTau else 0.0)
                  + (if o5 then w-3.0*w/checkTau else 0.0);
  tel
```

Comme on peut aisément le voir sur la Figure 8.1, la contribution d'un spike à l'état de la synapse à un temps donné dépend du temps écoulé depuis l'occurrence de ce spike. La variable booléenne $o1$ est vraie (vaut **true**) si et seulement si I_s était vraie au temps précédant le temps courant, ce qui signifie qu'un spike a été reçu à la synapse à cet instant. Similairement, $o2$ est vraie si I_s était vraie deux instants avant le temps courant, ce qui signifie qu'un spike a été reçu à la synapse il y a deux pas de temps, et ainsi de suite. Il faut ainsi définir autant

de variables locales `oN` que d'instants passés auxquels auraient pu être reçus des spikes qui influenceraient l'état courant de la synapse : `N` doit donc aller de 1 à `hatTau + checkTau - 1` (soit 5 dans cet exemple). Au-delà de cette fenêtre temporelle, les spikes n'ont plus d'influence sur l'état courant de la synapse (Chapitre 4, Section 4.2, Remarque Définition 12). L'état de la synapse `eta_s` est calculé en sommant toutes les contributions des spikes précédents (Figure 8.2). L'expression `eta_s = 0.0 -> (if o1 then w/hatTau else 0.0)` initialise `eta_s` à 0.0 puis, à chaque instant : si `o1` est vraie alors `eta_s` vaut `w/hatTau` (c'est-à-dire la contribution d'un spike reçu au temps précédent) sinon `eta_s` vaut 0.0.

Pour définir une synapse inhibitrice, il suffit d'affecter une valeur négative à `w`.

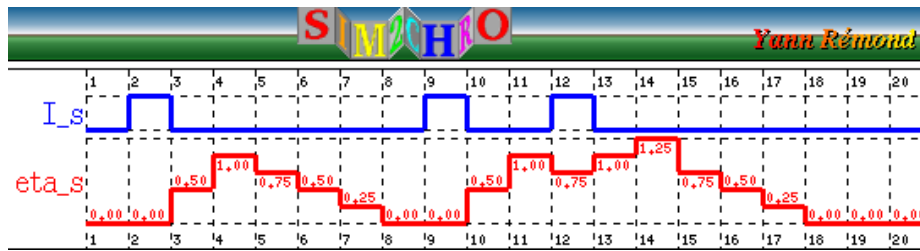


FIGURE 8.2 – Simulation du Nœud 1 par l'outil Luciole.

8.1.2 Encodage d'un compartiment

Un compartiment est un nœud qui calcule une sortie réelle `O` à partir d'un flot réel `I` (Figure 8.3). `I` représente l'état d'une synapse (comme défini dans la sous-section précédente). La sortie `O` correspond au flot de charges sortant du compartiment considéré à chaque instant. Notons que si on souhaitait implémenter un neurone non réduit, `I` pourrait également représenter la sortie d'un autre compartiment.

Pour rappel, un compartiment est un couple de paramètres (α, Δ) où α est l'atténuation et Δ est le temps de traversée du compartiment. Un nœud Lustre est requis pour tout paramètre Δ .

Prenons l'exemple d'un compartiment constitué de 3 sous-compartiments conformément à la discrétisation proposée dans le chapitre précédent (Figure 8.3). Dans cet exemple, on veut $\alpha = 0.125$ ce qui signifie qu'à l'issue de la traversée du compartiment, le signal est atténué par le facteur 0.125. Ceci nous amène à définir une variable `alphaSubC` qui correspond à l'atténuation propre à chaque sous-compartiment, qui vaut alors $\sqrt[3]{0.125} = 0.5$.

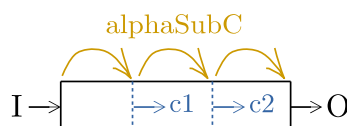


FIGURE 8.3 – Illustration du Nœud 2 qui encode un compartiment constitué de 3 sous-compartiments et pour lequel l'atténuation globale vaut alphaSubC^3 .

Nœud 2: Compartiment de paramètres $\alpha = 0.125$ et $\Delta = 3.\delta$ où δ est le pas de temps.

```
node compartiment (I : real) returns (O : real);
  var c1, c2, alphaSubC, init1, init2, init0 : real;
  let
    alphaSubC = 0.5;
    init1 = 0.0; init2 = 0.0; init0 = 0.0;
    c1 = init1 -> pre(I) * alphaSubC;
    c2 = init2 -> pre(c1) * alphaSubC;
    O = init0 -> pre(c2) * alphaSubC;
  tel
```

Les variables locales `c1` et `c2` correspondent respectivement à la valeur de la charge en sortie des deux premiers sous compartiments. La valeur de la charge en sortie du dernier sous-compartiments est finalement la valeur de la charge en sortie du compartiment dans son intégralité : il s'agit donc de la valeur de la sortie `O` (Figure 8.4). Puisque la durée de traversée de chaque sous-compartiment correspondant à un pas de temps, la sortie du compartiment se calcule simplement grâce à l'expression suivante : `O = init0 -> pre(c2) * alphaSubC`. La variable `O` est initialisée à une valeur arbitraire `init0` (ici 0.0) puis à chaque instant, elle prend la valeur de la charge en sortie de son sous-compartiment prédécesseur à l'instant précédent (`pre(c2)`), à laquelle est appliquée le facteur d'atténuation `alphaSubC`. La valeur de `c2` à l'instant précédent est elle-même calculée à partir de la valeur en sortie de son sous-compartiment prédécesseur à l'instant précédent (`pre(c1)`), et ainsi de suite. Lorsque l'on s'intéresse aux compartiments d'un neurone sous forme normalisé, aucune atténuation n'est appliquée, l'encodage du nœud peut alors se passer de la variable `alphaSubC`.

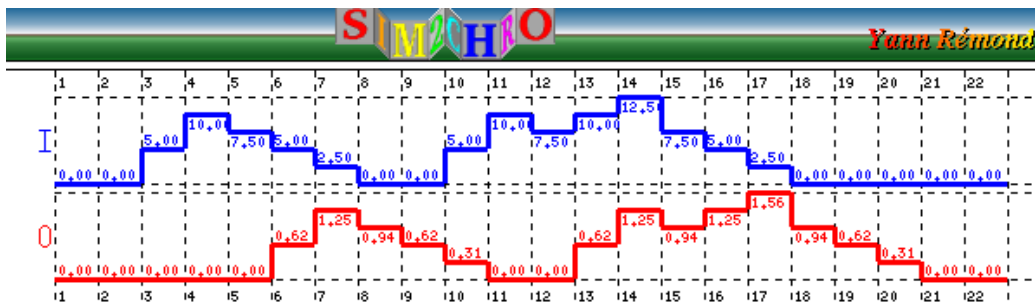


FIGURE 8.4 – Simulation du Nœud 2 par l'outil Luciole. La variable `I` correspond à la sortie de la synapse encodée par le Nœud 1 (mais avec un paramètre `w` multiplié par 10 pour une meilleure facilité de lecture). On affecte ici la valeur 0.0 aux variables `init0`, `init1` et `init2`.

8.1.3 Encodage d'un soma

Un soma est un nœud qui calcule une sortie booléenne `O` à partir d'un flot réel `F` (Figure 8.5). La sortie `O` constitue le signal de sortie du neurone. La variable `F` représente la quantité de charges entrant au soma à chaque instant : dans le contexte d'un neurone entier, il s'agit de

la somme des charges sortant des compartiments connectés au soma.

Nœud 3: Soma de paramètres $p_\beta = 10$, $\hat{p}_\beta = 10$, $e_\beta = 10$, $\hat{e}_\beta = 50$ et $\gamma = 0.1$.

```

node soma (F : real) returns (0 : bool);
  var eBeta, hat_eBeta, pBeta, hat_pBeta, gamma,
      varpBeta, dp, e, p, e0, p0 : real;
      PRA, PRR, normal : bool;
  let
    pBeta = 10.0;
    hat_pBeta = 10.0;
    eBeta = 10.0;
    hat_eBeta = 50.0;
    gamma = 0.1;
    e0 = 1000.0;
    p0 = 0.0;
    PRA = false -> if (pre(e) + 1 > eBeta) then false else true;
    PRR = false -> if (pre(e) + 1 > eBeta and pre(e) + 1 <= eBeta + hat_eBeta)
                    then true
                    else false;
    normal = true -> if (PRA or PRR) then false else true;
    varpBeta = pBeta + hat_pBeta -> if PRR
                                     then pre(varpBeta) - (hat_pBeta/hat_eBeta)
                                     else pBeta + hat_pBeta;

    dp = 0.0 -> F - gamma * pre(p);
    p = p0 -> if ((normal and pre(p) + dp < pBeta)
                 or (PRR and pre(p) + dp < varpBeta)
                 or (PRA))
              then pre(p) + dp
              else pre(p) + dp - pBeta;
    0 = false -> if ((normal and pre(p) + dp >= pBeta)
                   or (PRR and pre(p) + dp >= varpBeta))
                 then true
                 else false;
    e = e0 -> if (0) then 0.0 else pre(e) + 1.0;
  tel

```

Les variables locales `eBeta` et `hat_eBeta` correspondent à la durée des périodes réfractaires respectivement absolue et relative, `pBeta` est la valeur du seuil (hors période réfractaire), `hat_pBeta` est l'augmentation du seuil au sortir de la période réfractaire absolue, `gamma` est le facteur de fuite. Nous définissons également des variables additionnelles : `varpBeta` qui porte la valeur du seuil en période réfractaire relative, `dp` qui est la variation du potentiel au

soma à chaque instant, e qui porte la valeur du temps écoulé depuis l'émission du dernier spike et p qui porte la valeur du potentiel au soma. Les variables $p0$ et $e0$ portent les valeurs initiales de e et p .

À chaque instant, il faut tout d'abord connaître le statut du soma pour en déduire la valeur de son seuil. Les variables booléennes `PRA`, `PRR` et `normal` valent `true` lorsque le soma est respectivement en période réfractaire absolue, en période réfractaire relative ou en conditions normales (hors périodes réfractaires). Le soma est initialisé en conditions normales et le statut du soma dépend ensuite du temps écoulé depuis l'émission du dernier spike (e). On considère ici l'approximation linéaire de la fonction de borne. Ainsi, la valeur de `varpBeta` vaut `pBeta + hat_pBeta` au sortir de la période réfractaire absolue puis décroît régulièrement jusqu'à atteindre `pBeta` à la fin de la période réfractaire relative.

La variation dp du potentiel au soma est initialisée à 0.0 puis à chaque autre instant, elle correspond à la différence entre les charges qui entrent au soma (F) et celles qui fuient ($\gamma * pre(p)$). Pour calculer la valeur de p , dp est ajouté à chaque instant. Lorsque p atteint le seuil du soma (dont la valeur dépend du statut), la quantité `pBeta` lui est immédiatement retirée. À partir de la valeur de p , on peut alors définir le signal de sortie. Lorsque p atteint son seuil, un spike est émis sur la sortie c'est-à-dire que `0` vaut `true` à cet instant. La valeur du seuil est `pBeta` en conditions normales et `varpBeta` en période réfractaire relative. En période réfractaire absolue, il n'émettra jamais de spike, quelle que soit la valeur de p . Le temps écoulé depuis le dernier spike (e) est remis à 0.0 lorsque `0` vaut `true`, sinon il est incrémenté de 1.0 à chaque pas de temps.

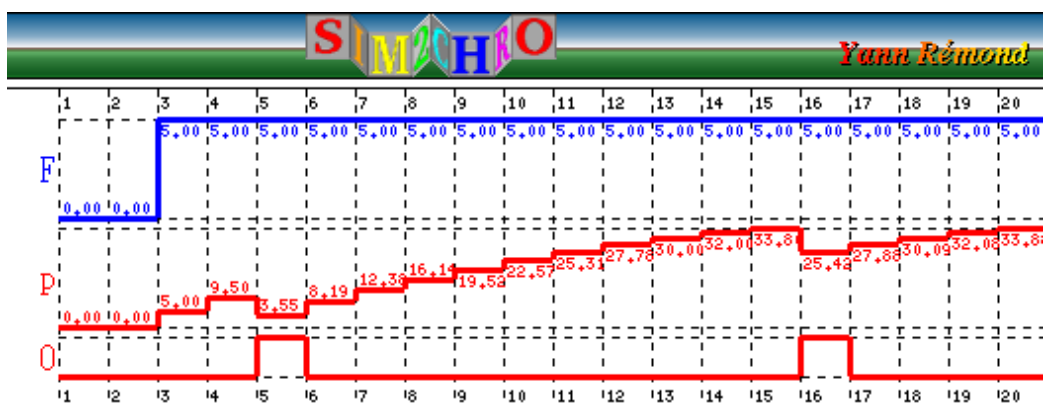


FIGURE 8.5 – Simulation du Nœud 3 par l'outil Luciole. La variable F représente la quantité de charges entrant au soma à chaque instant. La variable 0 correspond au signal de sortie du soma. Pour des raisons pédagogiques, on visualise également la valeur de p et on affecte la valeur 0.0 à $p0$ (le potentiel est à sa valeur de repos) et 1000.0 à $e0$ (le soma est initialisé hors période réfractaire). Un spike est émis sur 0 lorsque p atteint son seuil. On peut voir qu'aux derniers instants, la valeur de p est largement au-dessus du seuil, mais le neurone est en période réfractaire absolue ce qui explique que 0 soit à `false`.

8.1.4 Encodage d'un neurone

Une fois que les synapses et compartiments sont encodés sous forme de nœuds Lustre, il est possible d'implémenter un neurone complet. Nous choisissons de le faire en deux temps pour des raisons pratiques. Tout d'abord, nous définissons un nœud qui calcule la sortie de la forêt dendritique (Nœud 4). Ce flot de sortie réel est noté **F** en accord avec les définitions de notre cadre de modélisation (Chapitre 4, Section 4.2, Définition 16). Puis, dans un second nœud (Nœud 5), par l'intermédiaire du calcul de **F**, on calcule le signal de sortie du neurone à partir de son signal d'entrée.

Nœud 4: Dendrites d'un neurone réduit à 3 synapses.

```
node dendrites (I1, I2, I3 : bool) returns (F : real);
  var S1, S2, S3, C1, C2, C3, initF : real;
  let
    initF = 0.0;
    S1 = synapse1 (I1);
    S2 = synapse2 (I2);
    S3 = synapse3 (I3);
    C1 = compartiment1 (S1);
    C2 = compartiment2 (S2);
    C3 = compartiment3 (S3);
    F = initF -> C1 + C2 + C3;
  tel
```

Dans cet exemple (Nœud 4), on considère une forêt dendritique constituée de 3 compartiments (`compartiment1`, `compartiment2`, `compartiment3`), liés respectivement aux synapses (`synapse1`, `synapse2`, `synapse3`). Les variables réelles **S1**, **S2**, **S3** représentent l'état des synapses au cours du temps et servent d'entrées pour calculer la sortie des différents compartiments. La sortie du nœud **F** est simplement la somme des sorties des différents compartiments.

Nœud 5: Neurone réduit à 3 synapses.

```
node neurone (I1, I2, I3 : bool) returns (O : bool);
  var F : real;
  let
    F = dendrites (I1, I2, I3);
    O = soma (F);
  tel
```

Un neurone est finalement un nœud qui connecte la sortie d'une forêt dendritique (**F** = `dendrites (I1, I2, I3)`) à l'entrée d'un soma (**O** = `soma (F)`). De cette manière, le neurone est comme un système qui renvoie une sortie booléenne à partir d'entrées booléennes également. Toute l'intégration réelle est abstraite dans sa fonction d'entrée/sortie.

8.1.5 Encodage d'un réseau

Dans le contexte des assemblées de neurones, c'est l'échelle des réseaux qui est pertinente. La focalisation sur les entrées/sortie est en accord avec les approches expérimentales qui s'intéressent généralement à l'enregistrement des spikes.

Prenons l'exemple très simple d'un réseau à deux neurones `neurone1` et `neurone2` tel que l'entrée du `neurone2` est la sortie du `neurone1`. Le nœud `6` calcule la sortie `O2` du `neurone2` à partir de l'entrée `I1` du `neurone1`.

Nœud 6: Réseau à deux neurones.

```
node reseau (I1 : bool) returns (O2 : bool);
  var O1 : bool;
  let
    O1 = neurone1 (I1);
    O2 = neurone2 (O1);
  tel
```

Il s'agit là d'un microcircuit. On peut ensuite composer divers tels circuits, pour obtenir des réseaux bien plus grands et plus complexes.

8.2 Encodage et vérification de propriétés par model checking

L'un des avantages de Lustre est de permettre l'expression de propriétés globales dans le même langage que celui qui est utilisé pour implémenter le système à vérifier (Chapitre 2, Section 2.4). Les propriétés sont exprimées en Lustre dans des nœuds dédiés qui sont appelés *observateurs*. À chaque instant, l'observateur d'une propriété renvoie un booléen qui vaut `true` si la propriété est satisfaite, `false` sinon. Pour prouver que la propriété est satisfaite par le système, le model checking consiste alors à vérifier que l'observateur ne renvoie jamais `false`.

Nous choisissons d'utiliser le model checker *Kind2* (Chapitre 2, Section 2.4), qui a été montré comme étant plus efficace que les autres sur des systèmes relativement proches des nôtres [De Maria et al., 2016].

8.2.1 Preuve de concept

Pour preuve de concept, nous nous intéressons aux propriétés d'équivalence entre neurones (au sens des théorèmes que nous avons prouvés dans le Chapitre 5). Étant donné un neurone et sa forme réduite, nous allons prouver par model checking qu'ils ont la même sortie, pour des entrées particulières.

8.2. ENCODAGE ET VÉRIFICATION DE PROPRIÉTÉS PAR MODEL CHECKING

Nous prenons l'exemple simple illustré par la Figure 8.6. Puisqu'il s'agit d'un neurone et de sa forme réduite, ils ont tous deux le même soma (Chapitre 5, Section 5.1, Définition 21). Pour prouver qu'ils ont le même signal de sortie, il suffit donc de prouver que la quantité de charges entrant à leur soma est la même à chaque instant. Nous choisissons ainsi de faire la preuve d'équivalence de leur forêt dendritique. L'observateur de cette propriété est le Nœud 7 dans lequel `neurone1` et `neurone2` sont du type du Nœud 4. L'ensemble du code Lustre nécessaire pour prouver cette propriété est disponible en Annexe A.

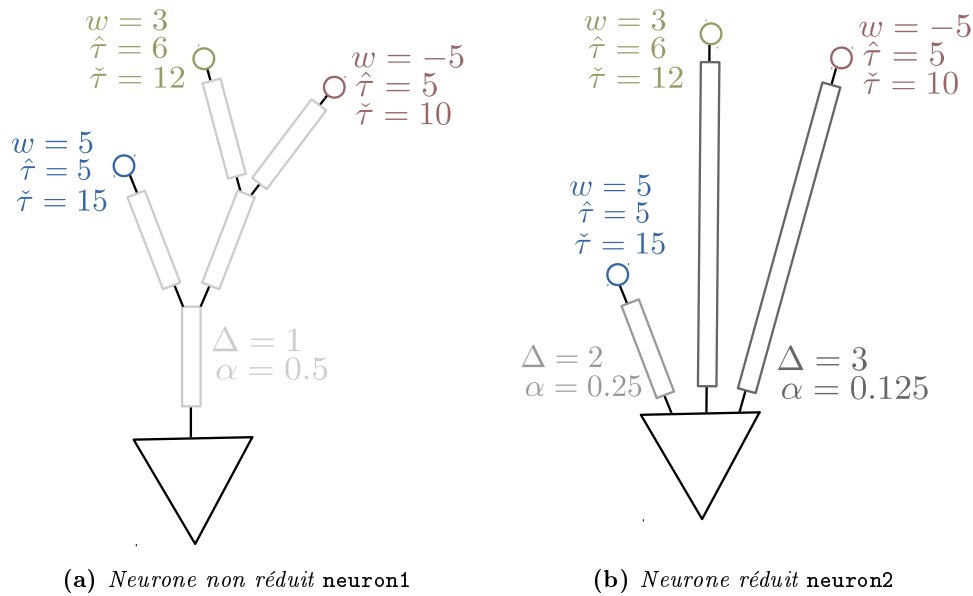


FIGURE 8.6 – Exemple d'un neurone `neurone1` et de sa forme réduite `neurone2` conforme à la Définition 21. Tous les paramètres temporels sont exprimés en multiples du pas de temps $\delta = 0.1s$.

Nœud 7: Observateur d'une propriété d'équivalence entre deux neurones.

```

node equivalence (gost : bool) returns (prop: bool);
  var I1, I2, I3 : bool;
  let
    I1 = true;
    I2 = true;
    I3 = false;
    prop = neurone1(I1, I2, I3) = neurone2(I1, I2, I3);
  tel

```

Le Nœud 7 possède une entrée factice (`gost`) et renvoie effectivement un booléen (`prop`). La variable `prop` prend la valeur `true` lorsque la sortie de `neurone1` est égale à la sortie de `neurone2` pour un même signal d'entrée (`I1`, `I2`, `I3`).

Pour des raisons de simplicité, nous fixons l'état initial des deux neurones à 0.0, ce qui est en accord avec la reconstruction d'état (Chapitre 5, Section 5.1, Définition 22). Nous commençons par des signaux d'entrée constants : soit toujours vrai, soit toujours faux.

CHAPITRE 8. PROTOTYPAGE AVEC LUSTRE

Puisque les neurones de notre exemple possèdent 3 synapses, il y a donc $2^3 = 8$ combinaisons possibles de signaux d'entrée. Pour chacune de ces combinaisons, le model checker prouve effectivement que la sortie des deux neurones est équivalente pour une même entrée. Les résultats sont rassemblés dans la Table 8.1.

Dans ces conditions, le model checker est remarquablement efficace. Chaque preuve, entièrement faite automatiquement, nécessite un temps de calcul allant de quelques secondes à quelques dizaines de secondes sur une machine standard.

| | | | | | | | | |
|--------------|-------|-------|-------|-------|-------|-------|-------|-------|
| I1 | false | true | false | true | false | true | true | false |
| I2 | true | false | false | true | true | false | true | false |
| I3 | false | false | true | false | true | true | true | false |
| Temps | 19,29 | 5,94 | 8,46 | 22,80 | 11,46 | 15,33 | 33,00 | 3,92 |

TABLE 8.1 – *Preuve d'équivalence par model checking. Pour toutes les combinaisons d'entrées présentées dans cette table, Kind2 prouve l'équivalence des sorties de la forêt dendritique des neurones illustrés en Figure 8.6. Le temps requis pour chaque preuve automatique est exprimé en secondes. L'expérience a été menée sur une machine Dell Intel Core i7-3770 CPU à 3.40GHz et 15,6 Go RAM.*

Puisque les neurones traitent des nombres réels, les erreurs d'arrondis dans les calculs empêchent de définir la propriété `prop` (Nœud 7) par la simple égalité `prop = neurone1(I1, I2, I3) = neurone2(I1, I2, I3)`. Pour pallier ce problème, l'observateur doit être implémenté à la manière du Nœud 8. Cependant, dans le cas de notre exemple (Figure 8.6), la preuve par model checking aboutit même avec l'emploi de la simple égalité. Ceci signifie que, s'il y en a, `neurone1` et `neurone2` font les mêmes approximations.

Nœud 8: Observateur d'une propriété d'équivalence (à epsilon près).

```
node equivalence (gost : bool, epsilon: real) returns (prop: bool);
  var I1, I2, I3 : bool;
  let
    I1 = true;
    I2 = true;
    I3 = false;
    prop = abs(neurone1(I1, I2, I3) - neurone2(I1, I2, I3)) < epsilon;
  tel
```

Les résultats présentés dans la Table 8.1 concernent des preuves faites sur la forêt dendritique des neurones de la Figure 8.6. Lorsqu'on reproduit l'expérience, cette fois-ci en prenant en compte le soma (code en Annexe B), le temps requis pour chaque preuve est en moyenne multiplié par près de 100 (Table 8.2). Ce temps de calcul est plus réaliste, car la présence du soma est généralement indispensable pour étudier le comportement des neurones. Contrairement à la preuve d'équivalence entre un neurone et sa forme réduite, la prise en compte du soma est nécessaire pour prouver l'équivalence entre un neurone et sa forme normalisée,

8.2. ENCODAGE ET VÉRIFICATION DE PROPRIÉTÉS PAR MODEL CHECKING

par exemple.

| | | | | | | | | |
|--------------|---------|--------|---------|--------|---------|--------|--------|---------|
| I1 | false | true | false | true | false | true | true | false |
| I2 | true | false | false | true | true | false | true | false |
| I3 | false | false | true | false | true | true | true | false |
| Temps | 1048,47 | 201,08 | 1299,02 | 412,59 | 1217,64 | 290,36 | 254,22 | 1110,84 |

TABLE 8.2 – *Preuve d'équivalence par model checking (avec soma). Pour toutes les combinaisons d'entrées présentées dans cette table, Kind2 prouve l'équivalence des sorties booléennes des neurones illustrés en Figure 8.6. Le temps requis pour chaque preuve automatique est exprimé en secondes. L'expérience a été menée sur une machine Dell Intel Core i7-3770 CPU à 3.40GHz et 15,6 Go RAM.*

Les entrées constantes considérées jusqu'ici n'ont aucune plausibilité biologique. En particulier, un signal ne peut être toujours « vrai » puisque deux spikes doivent au moins être séparés par la durée de la période réfractaire. Nous avons donc également prouvé par model checking la propriété d'équivalence pour d'autres signaux d'entrée plus élaborés, définis par des fréquences de spikes arbitraires. Par exemple, nous avons défini les flots **I1**, **I2** et **I3** tels qu'ils valent **true** respectivement à un temps sur 50, à un temps sur 10 et à un temps sur 20 (code en Annexe C). Dans ce cas et sans considérer le soma, le model checker parvient à prouver l'équivalence, mais en un temps déraisonnable : il lui faut alors près de 15500 secondes, soit plusieurs heures pour fournir un résultat.

8.2.2 Limitations

Dans ce chapitre, nous avons montré qu'il est possible de prouver automatiquement des propriétés en se basant sur le cadre de modélisation que nous avons défini dans les Chapitres 4 et 5. Il s'agit là d'une simple preuve de concept.

Dans le Chapitre 5, nous avons mathématiquement prouvé qu'un neurone et sa forme réduite partagent le même signal de sortie, quel que soit leur signal d'entrée commun. Nous avons ici fait la preuve de cette propriété par model checking pour des cas particuliers de neurones et de signaux d'entrée.

En prenant l'exemple simple d'un neurone à 3 synapses (Figure 8.6), nous avons automatiquement prouvé grâce au model checker Kind2, que le processus de réduction préserve effectivement le signal de sortie en réponse à des signaux d'entrée constants. En vue d'améliorer la plausibilité biologique des signaux d'entrée, nous avons également prouvé cette propriété pour des signaux d'entrée ayant une fréquence arbitrairement fixée. Nous avons alors atteint les limites de Kind2 sur une machine standard.

La faisabilité de ces preuves sans intervention humaine est tout de même encourageante malgré la relative simplicité des propriétés. Pour progresser vers des propriétés plus complexes, on pourrait simplement augmenter les ressources de calcul, mais cette stratégie n'est pas celle que nous souhaitons privilégier puisque notre objectif à long terme est d'étudier

facilement de grands réseaux de neurones. On pourrait également simplifier notre modèle. Cependant, nous perdrons en plausibilité biologique et le modèle que nous avons défini possède déjà les propriétés que nous jugeons minimales dans le contexte des assemblées. Enfin, on pourrait s’orienter vers des alternatives à Kind2 : en testant d’autres outils de model checking ou en optant pour d’autres techniques de vérification éventuellement développées spécifiquement pour notre système.

Par ailleurs, au-delà de la plausibilité des signaux d’entrée, il faut également considérer la plausibilité des neurones implémentés. En particulier, un neurone biologique possède près de 10000 synapses ce qui augmente largement le nombre de paramètres. Pour limiter cette complexité, on pourrait envisager un niveau d’abstraction supérieur en rassemblant des synapses pour former des clusters par exemple.

Si l’on s’éloigne de l’approche formelle, on peut également envisager la simulation. En Lustre et grâce à notre étape de discrétisation, la simulation est particulièrement efficace. De plus, puisque notre modèle est déterministe, il n’est pas nécessaire de répéter les simulations pour un même jeu de paramètres. Cependant, Lustre est un langage qui manque de modularité dans le sens où beaucoup de code doit être répété : par exemple, l’absence de boucle oblige à définir un nœud pour chaque type de synapse. Pour pallier cette limitation, nous avons implémenté un programme (en Java) permettant de générer du code Lustre (voir Annexe D pour l’exemple de la synapse).

8.3 Exemple de simulation du modèle Lustre

Par un exemple simple de simulation, nous mettons en évidence ici l’importance de prendre en compte les dendrites dans les modèles, en vue d’étudier les assemblées de neurones.

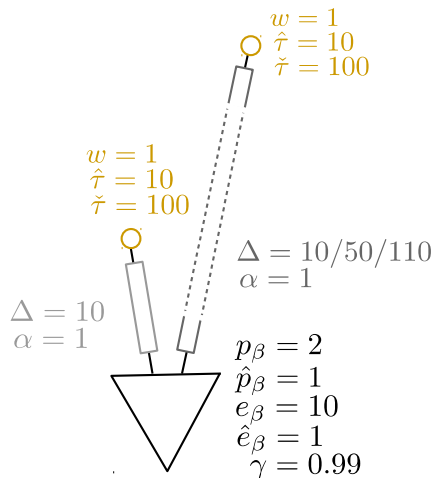


FIGURE 8.7 – Exemple de neurone réduit à 2 synapses identiques. Le temps de traversée du compartiment en pointillés peut valoir 10, 50 ou 110 pas de temps. Tous les paramètres temporels sont exprimés en multiples du pas de temps $\delta = 0.1s$.

8.3. EXEMPLE DE SIMULATION DU MODÈLE LUSTRE

Nous implémentons, en Lustre, le neurone réduit illustré par la Figure 8.7. Les paramètres temporels sont inspirés de la réalité biologique, en particulier les paramètres caractéristiques des synapses [Miles and Wong, 1986] [Magee and Cook, 2000] [Shepherd, 2004] et la durée de la période réfractaire absolue [Biedermann-Thorson et al., 2013]. Dans cet exemple, la période réfractaire relative est très courte [Koch, 2004] et la fuite au soma est particulièrement importante. Ce neurone ne possédant que deux synapses, il est nécessaire de choisir un seuil très faible pour qu’il soit fonctionnel. Au regard de ces paramètres, au minimum deux spikes reçus sont nécessaires pour déclencher l’émission d’un spike par le neurone.

Dans cette expérience, nous fournissons un signal d’entrée identique aux deux synapses du neurone. Ce signal correspond à un spike tous les 100 pas de temps. Puisque le facteur de fuite est fort, deux spikes successifs reçus sur une même synapse ne suffisent pas à provoquer l’émission d’un spike. Les spikes doivent ainsi être coordonnés sur les deux synapses pour atteindre le seuil.

La simulation, menée sur 1000 pas, est répétée pour différents temps de traversée du compartiment représenté en pointillés sur la Figure 8.7. Le temps de traversée vaut soit 10, soit 50, soit 110 pas de temps δ (Figure 8.8). Ces délais sont représentatifs de la variabilité qui peut être observée expérimentalement [Stuart et al., 2016].

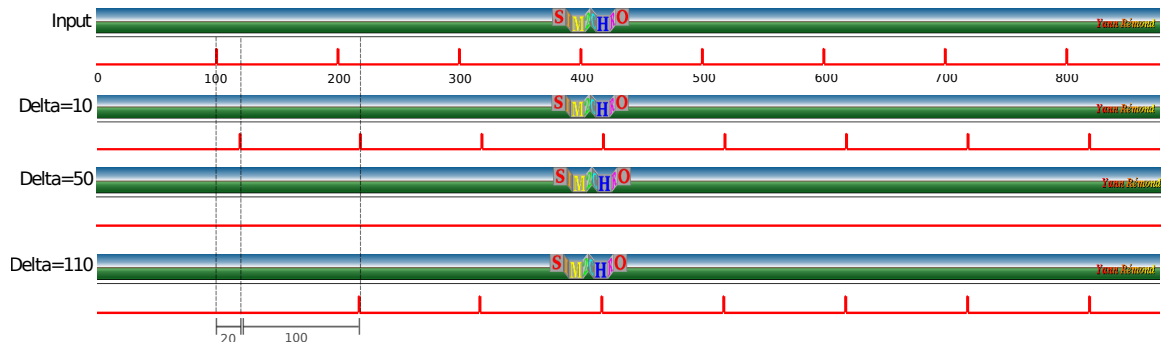


FIGURE 8.8 – Simulation du neurone illustré en Figure 8.7 avec l’outil Luciole. Les lignes correspondent respectivement au signal d’entrée commun reçu par les deux synapses du neurone (Input) et au signal de sortie du neurone en réponse à ce signal d’entrée pour différents délais de traversée (Delta) du compartiment représenté en pointillés sur la Figure 8.7. Les temps sont exprimés en multiples du pas de temps $\delta = 0.1s$. La valeur du potentiel au soma est partiellement illustrée en Annexe E.

Lorsque les deux compartiments ont le même temps de traversée (10δ), les spikes reçus aux synapses sont intégrés aux mêmes temps au niveau du soma, permettant d’atteindre le seuil et ainsi d’émettre un spike. Dans ces conditions, le neurone agit comme un « retardateur » qui reproduit le signal d’entrée sur sa sortie avec un décalage temporel provenant du temps de traversée des dendrites (Figure 8.8, ligne 2).

Lorsque le temps de traversée du compartiment représenté en pointillés sur la Figure 8.7 vaut 50δ , il se produit un décalage temporel dans l’intégration somatique des spikes reçus. Effectivement, les spikes sont reçus aux deux synapses en même temps, mais leur arrivée au

soma, qui dépend du temps de traversée des compartiments, se trouve ainsi désynchronisée. Puisqu'un spike ne suffit pas à atteindre le seuil, la sortie du neurone est constamment nulle dans ces conditions (Figure 8.8, ligne 3).

L'augmentation du temps de traversée n'est pas le seul paramètre responsable du changement de fonction du neurone. Lorsque le temps de traversée du compartiment représenté en pointillés vaut 110δ , la simulation montre que le neurone produit une sortie du même type que dans le cas où le temps de traversée vaut 10δ (Figure 8.8, ligne 4). Ceci s'explique facilement : lorsqu'un spike reçu à la première synapse est intégré au soma (soit 10 pas de temps après sa réception), il s'ajoute à l'effet du spike reçu 110 pas de temps plus tôt à l'autre synapse. Ceci permet au neurone d'atteindre son seuil et d'émettre un spike.

Le résultat de cette simple simulation montre bien l'influence des délais apportés par les dendrites sur la fonction d'entrée/sortie du neurone. En particulier, le *rapport entre les différents délais* semble avoir une importance cruciale. Les délais (en combinaison avec la temporalité des signaux d'entrée) sont donc essentiels pour la coordination temporelle et par conséquent pour la précision des temps d'émission des spikes.

Bien que souvent négligées, les considérations morphologiques apparaissent déterminantes dans le contexte des assemblées et plus largement dans les processus de synchronisation (Chapitre 1, Section 1.3). Notre modèle, en intégrant des dendrites, semble plus pertinent pour étudier ces structures que les modèles abstraits existants qui ne tiennent pas compte des délais de transmission (Chapitre 3, Section 3.1).

CHAPITRE 9

INTRODUCTION DE LA PLASTICITÉ SYNAPTIQUE

Dans ce chapitre, nous allons exploiter le formalisme de base présenté dans les chapitres précédents et l'enrichir pour modéliser en sus la plasticité synaptique. Comme on va le voir, le « supplément théorique » que cela requiert n'est pas volumineux et cette relative facilité d'enrichissement nous semble un indice de bonne structuration du formalisme de base. Nous n'avons en fait pour objectif dans ce chapitre que de démontrer la souplesse de notre formalisme et nous n'irons pas, par exemple, jusqu'à l'implémentation en Lustre de la plasticité synaptique (*a priori* sans difficulté majeure) ou l'étude de morphismes adaptés en théorie des catégories pour l'étude de propriétés algébriques globales (un obstacle majeur venant du fait que la plasticité synaptique dépend de l'état du soma et donc de l'intégrale qu'il implique).

Nous proposons ici une forme de plasticité particulière, basée sur la *STDP* (pour *Spike-Timing-Dependent Plasticity*) [Bi and Poo, 1998] (Chapitre 1, Section 1.2). Dans une première section, nous donnons une vue d'ensemble de l'introduction de la plasticité dans notre modèle. Dans la suivante, nous formalisons la dynamique du neurone en intégrant la plasticité. Enfin, nous discutons cette extension.

9.1 Vue d'ensemble

La mise en place des assemblées de neurones reposerait sur des mécanismes de plasticité synaptique (Chapitre 1, Section 1.3). Pour étudier *in silico* ce type de phénomène, la plasticité synaptique doit donc également être modélisée au niveau des neurones individuels. Ceci signifie que l'efficacité des synapses des modèles doit varier au cours du temps, en fonction de la dynamique du neurone et des signaux qu'il reçoit.

Dans un modèle purement linéaire, l'efficacité d'une synapse s est définie par la valeur de son paramètre w_s , qui représente la valeur maximale atteinte par sa fonction synaptique en

CHAPITRE 9. INTRODUCTION DE LA PLASTICITÉ SYNAPTIQUE

réponse à un spike (Chapitre 4, Section 4.1, Définition 3). Dans notre formalisme algébrique, le paramètre w_s (le poids de s) est statique. Dans cette section, nous développons les idées principales permettant de rendre ce paramètre dynamique.

Afin d'introduire de la plasticité synaptique, nous définissons dans ce chapitre une *dynamique enrichie* \bar{d} (Définition 35 section suivante) qui dépend d'un signal d'entrée I et associe à chaque temps un *état enrichi* $E = (N, \eta)$ (Définition 34 section suivante) où N est un neurone au sens de la Définition 7 et η est un état au sens de la Définition 8 (Section 4.1, Chapitre 4). La progression des états η est directement calculée en suivant la Définition 16 de la dynamique du neurone (Chapitre 4, Section 4.2). Quant à la succession des neurones N , c'est elle qui porte les éventuelles modifications du poids de certaines synapses, et elle sera cadencée par les spikes *émis* sur la sortie au cours du temps.

Finalement, plutôt que de modifier les poids des synapses d'un neurone au cours du temps (ce qui impliquerait de redéfinir des paramètres du cadre de modélisation présenté dans le Chapitre 4), nous choisissons de « créer », à différents temps, un nouveau neurone qui succède au précédent : les poids des synapses peuvent alors varier d'un neurone à son successeur. Cette manipulation permet d'introduire de la plasticité synaptique à notre cadre de modélisation, sans aucune modification du formalisme initial.

Dans cette thèse, nous accordons une importance particulière aux temps précis des spikes. C'est donc très naturellement que nous choisissons l'approche *STDP* pour modifier le poids des synapses (Chapitre 1, Section 1.2). La *STDP* potentialise une synapse lorsque le neurone post-synaptique émet un spike après le neurone pré-synaptique, et déprécie une synapse dans le cas contraire. Ce type de plasticité prend en compte non seulement la corrélation, mais aussi la causalité entre les spikes. L'idée est de renforcer une synapse lorsque le neurone pré-synaptique est susceptible d'avoir contribué à l'émission d'un spike par le neurone post-synaptique.

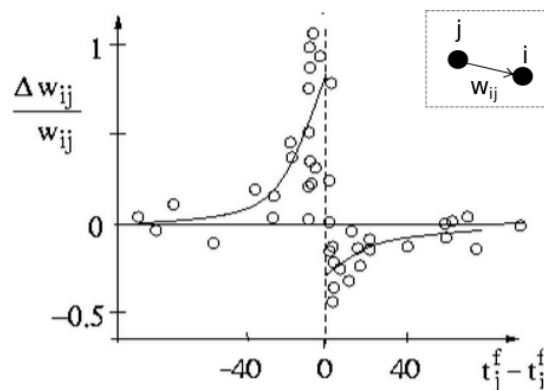


FIGURE 9.1 – Fonction basique de *STDP* qui représente la modification du poids w_{ij} de la synapse entre un neurone pré-synaptique j et un neurone post-synaptique i . La partie gauche de la courbe correspond à la potentialisation, lorsque j émet un spike avant i . La partie droite de la courbe correspond à la dépression, lorsque j émet un spike après i . Le délai $t_j^f - t_i^f$ entre l'émission des spikes par i et j , est exprimé en ms. *Figure extraite de [Sjöström and Gerstner, 2010].*

Dans la version basique de la STDP [Sjöström and Gerstner, 2010], la variation globale du poids $\Delta_{w_{ij}}$ de la synapse liant le neurone pré-synaptique j au neurone post-synaptique i suit la loi :

$$\Delta_{w_{ij}} = \sum_{f=1}^N \sum_{n=1}^N W(t_j^f - t_i^n)$$

où t_i^n est le temps d'émission du n^{ime} spike par le neurone i , t_j^f est le temps d'émission du f^{ime} spike par le neurone j et $W(x)$ est la fonction de STDP.

Il est également possible de calculer la valeur du poids synaptique w_{ij} de la synapse liant le neurone pré-synaptique j au neurone post-synaptique i , à un temps $t + h$ à partir de sa valeur à un temps t [Guyonneau et al., 2005] : $w_{ij}(t + h) = w_{ij}(t) + W(t_j^f - t_i^f)$.

La Figure 9.1 représente la fonction classique de STDP, qui correspond aux mesures expérimentales sur des neurones d'hippocampe en culture [Bi and Poo, 1998]. Elle est constituée de deux parties distinctes, l'une correspondant à la diminution du poids synaptique et l'autre correspondant à son augmentation. La courbe représentative de cette fonction est « calée » sur zéro, c'est-à-dire que l'augmentation du poids concerne des délais négatifs tandis que la diminution concerne des délais positifs. Ceci signifie que plus le temps du spike pré-synaptique est proche du temps du spike post-synaptique, plus la variation du poids synaptique est importante.

Dans notre formalisme, si l'on veut effectivement potentialiser les synapses qui ont contribué à l'émission d'un spike, ce type de fonction STDP n'est pas pleinement pertinent tel quel. Par exemple, prenons un neurone réduit N et une synapse s liée au soma de N par un compartiment $c = (\alpha, \Delta)$. Si N reçoit un spike sur s « juste avant d'émettre un spike », il est peu probable que le spike pré-synaptique ait contribué au spike post-synaptique. Effectivement, si le délai entre les deux spikes est inférieur à Δ , l'effet du spike pré-synaptique « n'aura pas eu le temps » d'être intégré au soma avant l'atteinte du seuil. Nous devons donc prendre en compte les délais apportés par les dendrites dans l'implémentation de la plasticité synaptique. Nous choisissons de « décaler » la fonction STDP classique (Figure 9.1), pour la centrer non pas sur zéro, mais sur un temps compatible avec le délai de traversée des compartiments. Nous appelons dès lors cette fonction, une fonction de *d-STDP* (pour *delayed-STDP*), laquelle doit donc être paramétrée de façon spécifique pour chacune des synapses.

9.2 Formalisation

Notre cadre de modélisation, tel qu'il est défini, nous permet de connaître la fenêtre temporelle durant laquelle un spike pré-synaptique peut contribuer à l'émission d'un spike post-synaptique (Figure 9.2). Pour contribuer à l'émission d'un spike à un temps t , la variation de charge produite par le spike pré-synaptique doit, au moins, avoir atteint le soma à ce temps-ci. Il faut donc que le spike pré-synaptique ait été reçu à la synapse, au plus tard, au

CHAPITRE 9. INTRODUCTION DE LA PLASTICITÉ SYNAPTIQUE

temps $t - \Sigma^{s \rightarrow \nabla}$, où $\Sigma^{s \rightarrow \nabla}$ est le temps de traversée des branches de l'arbre dendritique, de la synapse s au soma ∇ . Puisque ce délai $\Sigma^{s \rightarrow \nabla}$ varie d'une synapse à l'autre, nous choisissons de définir une fonction de d-STDP pour chaque synapse (Définition 33 ci-après). On connaît également le temps à partir duquel le spike pré-synaptique aura été reçu trop tôt pour contribuer à l'atteinte du seuil (Chapitre 4, Section 4.2, Remarque Définition 12) : pour contribuer à l'émission d'un spike post-synaptique à un temps t , un spike pré-synaptique doit être reçu à une synapse s , dans la fenêtre temporelle $]t - (\Sigma^{s \rightarrow \nabla} + TC_s), t - \Sigma^{s \rightarrow \nabla}[$ où $TC_s = \hat{\tau}_s + \check{\tau}_s$ est le temps de contribution d'un spike reçu sur s . C'est donc sous cette condition que nous choisissons de renforcer l'efficacité de s .

Définition 33. [Fonction de d-STDP]

Soient un neurone N et une synapse $s \in Sy(N)$. Une fonction de d-STDP pour s est une fonction $W_s : \mathbb{R} \rightarrow \mathbb{R}$ telle que :

- W_s est continue et croissante sur $[-(\Sigma^{s \rightarrow \nabla} + TC_s), -\Sigma^{s \rightarrow \nabla}[$ et sur $[-\Sigma^{s \rightarrow \nabla}, T]$ où T est un nombre réel strictement supérieur à $-\Sigma^{s \rightarrow \nabla}$;
- il existe deux nombres réels W_s^+ et W_s^- tels que :
 - $\lim_{x \rightarrow -\Sigma^{s \rightarrow \nabla}^-} W_s(x) = W_s^+$
 - $W_s(-\Sigma^{s \rightarrow \nabla}) = W_s^-$;
- W_s est nulle pour tout x tel que $x \leq -(\Sigma^{s \rightarrow \nabla} + TC_s)$ ou $x \geq T$.

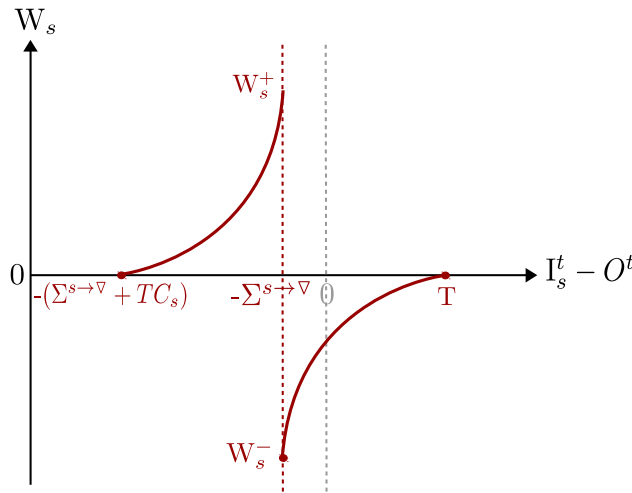


FIGURE 9.2 – Fonction de d-STDP (Définition 33) d'une synapse s , liée au soma par un arbre dendritique dont le temps de parcours jusqu'au soma est $\Sigma^{s \rightarrow \nabla}$. Nous prenons ici l'exemple d'une fonction symétrique. La variation maximale du poids de s est représentée par les paramètres W_s^+ et W_s^- .

D'après les observations expérimentales, il existe différentes formes de STDP [Feldman, 2012]. La variation provient surtout de l'asymétrie possible entre la potentialisation et la dépression des synapses. La forme de STDP pourrait dépendre du type des synapses et également de leur placement sur les dendrites [Froemke et al., 2010] [Feldman, 2012]. Pour cette raison,

nous laissons libre la forme exacte de la fonction de d-STDP, en imposant uniquement le sens de variation.

Les fonctions W_s de d-STDP permettront de calculer le poids des synapses, pour chaque état enrichi de la dynamique enrichie (Définitions 34 et 35 plus loin). Pour simplifier les définitions suivantes, on note préalablement \bar{N} le neurone obtenu par modification des poids synaptiques d'un neurone N , selon la loi de d-STDP :

Notation 4. *Étant donné un neurone N , un état initial $\eta_0 \in \zeta_N$, un signal d'entrée I , et une famille de fonction de d-STDP $W = \{W_s\}_{s \in Sy(N)}$, soit $t = \inf\{t' \in \mathbb{R}^+ \mid O_{N,I}(t') = 1\}$, on note \bar{N} le neurone tel que :*

- $\nabla(\bar{N}) = \nabla(N)$;
- $Co(\bar{N}) = Co(N)$;
- $Sy(\bar{N})$ contient les mêmes symboles que $Sy(N)$ et pour toute synapse $s \in Sy(N)$, on note $X_s^t = \{x \in [-(\Sigma^{s \rightarrow \nabla} + TC_s), -\Sigma^{s \rightarrow \nabla}] \mid I_s(t+x) = 1\}$. Dès lors, la fonction synaptique de s dans \bar{N} est définie par :

$$\sigma_{\bar{s}} = \sigma_s \times \left(1 + \frac{\sum_{x \in X_s^t} W_s(x)}{w_s} \right)$$

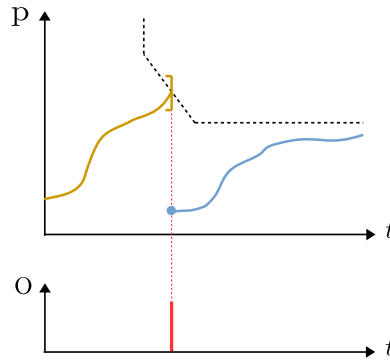


FIGURE 9.3 – *Dynamique du soma d'un neurone N . Partant d'une condition initiale, le potentiel de soma p suit sa dérivée. Lorsque p atteint la fonction de borne (ici linéaire par morceaux) qui représente le seuil, un spike est émis sur la sortie (O , en bas de la figure), mettant à jour p . Puis p continue d'évoluer suivant sa dérivée, partant d'une nouvelle condition initiale. Par définition (Chapitre 4, Section 4.2, Lemme 3), p n'atteint jamais le seuil. Lorsque la valeur de p (calculée en prenant en compte les charges entrant dans le soma et celles qui fuient) « devrait » dépasser le seuil, le neurone émet un spike et la valeur de p est immédiatement mise à jour. Cette modélisation nous permet de changer le poids des synapses en conservant le même état, sans réémettre « le spike » à l'origine de la plasticité.*

La dynamique enrichie est une succession d'états enrichis, c'est-à-dire une succession de neurones avec leurs états associés (Définitions 34 et 35 ci-après). Elle commence avec un état enrichi initial, qui suit sa dynamique habituelle (Chapitre 4, Section 4.2) tant que le

CHAPITRE 9. INTRODUCTION DE LA PLASTICITÉ SYNAPTIQUE

neurone n'émet pas de spike. Lorsque le neurone initial émet un spike, le neurone change au temps suivant et l'état reste le même (Figure 9.3). À partir de ce nouvel état initial, le neurone (modifié) suit de nouveau sa dynamique habituelle. De manière plus détaillée : à tout temps t (non nul) tel qu'un spike est émis sur la sortie, le neurone reste le même qu'à ε plus tôt (où ε est inférieur à la période réfractaire, ce qui signifie qu'aucun spike n'a pu être émis entre t et $t - \varepsilon$, modifiant le neurone), puis les poids synaptiques sont éventuellement modifiés par la loi de d-STDP, générant immédiatement un nouveau neurone.

Remarque: *Lorsqu'un neurone émet un spike sur sa sortie, le poids des synapses de ce neurone peut changer. À cet instant même, l'état du neurone n'est potentiellement plus en continuité avec le signal d'entrée (Chapitre 4, Section 4.2, Définition 13). Ceci n'est pas un problème puisque toutes les définitions des chapitres précédents restent vraies même sans la condition de continuité. Imposer la continuité n'est pertinent d'un point de vue biologique que lorsque les poids synaptiques du neurone ne changent pas au cours du temps.*

Définition 34. [État enrichi d'un neurone]

Un état enrichi est un couple $E = (N, \eta)$ où :

- $N \in \mathcal{N}$ est un neurone ;
- $\eta \in \zeta_N$ est un état du neurone N .

On note \mathcal{E} , l'ensemble des états enrichis.

Définition 35. [Dynamique enrichie d'un neurone]

Soient $E = (N^0, \eta^0)$ un état enrichi et I un signal d'entrée pour N^0 . La dynamique enrichie et le signal de sortie enrichi de N^0 conformément à I avec η^0 comme état initial, sont les fonctions $\bar{d}_{E,I} : \mathbb{R}^+ \rightarrow \mathcal{E}$ et $\bar{O}_{E,I} : \mathbb{R}^+ \rightarrow \{0, 1\}$ définies inductivement comme suit, où $E^t = (N^t, \eta^t)$ dénote $\bar{d}_{E,I}(t)$.

- si $O_{N^0,I}(0) = 0$ alors, pour tout $\delta \in \mathbb{R}^+$ tel que $O_{N^0,I}([0, \delta]) = \{0\}$, on a :
 - $\bar{O}_{E,I}(\delta) = 0$;
 - $E^\delta = (N^\delta, \eta^\delta)$ où $\eta^\delta = d_{N^0,I}(\delta)$;
- pour tout $t \in \mathbb{R}^+$ tel que $O_{N^t,I}(t) = 1$, en notant I^t la fonction $I^t(h) = I(h+t)$, on a $\bar{O}_{E,I}(t) = 1$ et :
 - si $t = 0$ alors $E^t = E$;
 - si $t \neq 0$ alors $E^t = (N^t, \eta^t)$ où $N^t = N^{t-\varepsilon}$ avec $0 < \varepsilon < e_\beta$ (e_β est la période réfractaire absolue de N^0 , Définitions 5 et 6 du Chapitre 4) et $\eta^t = d_{N^t,I^t}(t)$ où $t' = \max\{t'' < t \mid \bar{O}_{E,I}(t'') = 1\}$. De plus, pour tout $\delta \in \mathbb{R}^+$ tel que $O_{N^t,I}([t, t+\delta]) = \{0\}$, on a :
 - * $\bar{O}_{E,I}(t+\delta) = 0$;
 - * $E^{t+\delta} = (\bar{N}^t, \eta^{t+\delta})$ où $\eta^{t+\delta} = d_{\bar{N}^t,I^t}(\delta)$ avec η^t pour état initial.

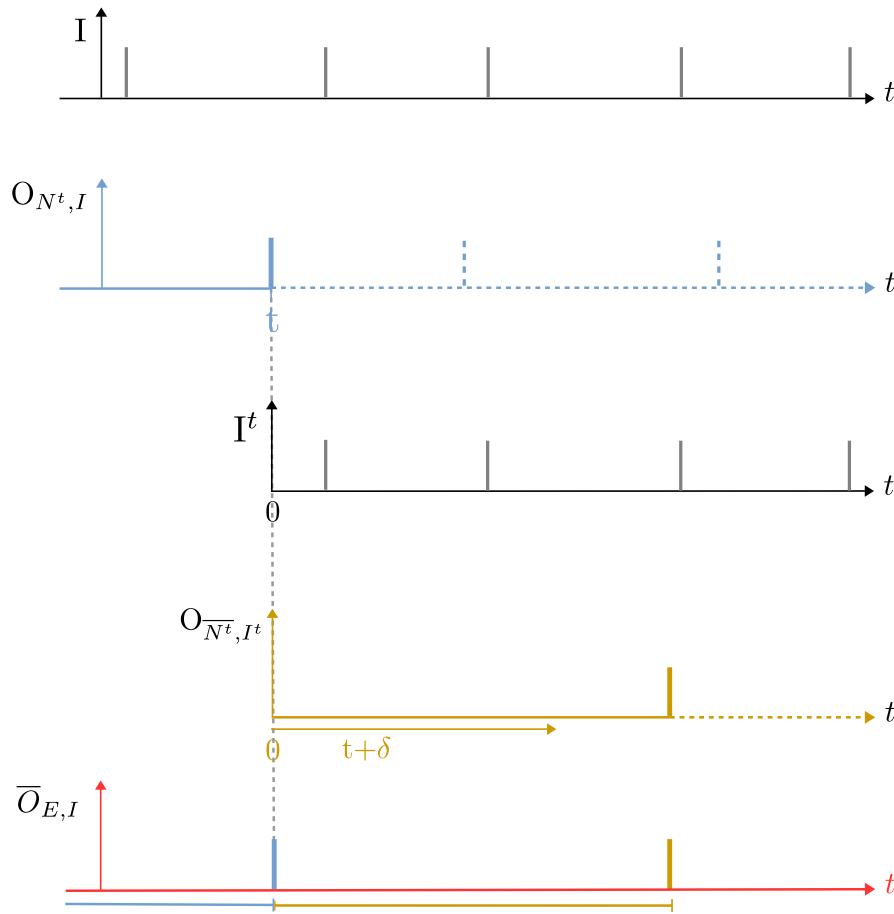


FIGURE 9.4 – Dynamique enrichie d’un neurone ayant pour état initial E , en fonction d’un signal d’entrée I . La sortie du neurone est notée $\bar{O}_{E,I}$. On suppose ici que $N^0 = N^t$. Lorsque le neurone N^t émet un spike sur sa sortie $O_{N^t, I}$, le poids de ses synapses peut éventuellement changer pour donner le neurone \bar{N}^t . C’est alors la dynamique de ce neurone \bar{N}^t qui définit la sortie $\bar{O}_{E,I}$, jusqu’à l’émission d’un spike pouvant éventuellement faire changer le poids de ses synapses, et ainsi de suite.

9.3 Discussion

Dans ce chapitre, nous avons étendu notre cadre de modélisation afin d’intégrer la plasticité synaptique. Il s’agit d’un phénomène crucial dans la formation des assemblées de neurones et plus généralement dans les mécanismes d’apprentissage.

La définition inductive de la dynamique enrichie que nous proposons semble *a priori* bien adaptée à une implémentation discrète. Concernant la discrétisation de notre cadre de modélisation, la plasticité synaptique implique de tester la sortie du neurone et de la comparer aux spikes d’entrée, à chaque pas de temps. La démarche de discrétisation présentée dans le Chapitre 7 impose que les spikes soient produits uniquement à ces pas de temps. L’introduction de la plasticité n’induit donc pas d’imprécision supplémentaire dans le calcul des délais entre les spikes pré- et post-synaptiques.

Pour alléger l’implémentation, il serait judicieux de n’implémenter qu’une seule fonction

CHAPITRE 9. INTRODUCTION DE LA PLASTICITÉ SYNAPTIQUE

W de STDP partagée par toutes les synapses, aux translations près de $-\Sigma^{s \rightarrow v}$ à partir d'une forme de référence calée sur zéro, plutôt que la famille $\{W_s\}$ qui représente une forme différente pour chaque synapse. Dans ce cas, le délai cumulé entre chaque synapse et le soma pourrait être pris en compte directement lors de la mise à jour des poids synaptiques. Par exemple, si le délai entre un spike pré-synaptique reçu à une synapse s et un spike post-synaptique vaut δ , on appliquera sur s la variation de poids correspondant au délai $\delta - \Sigma^{s \rightarrow v}$ pour intégrer le délai propre à la d-STDP.

Comme déjà mentionné en début de chapitre, nous n'avons pas poussé plus loin le volet « simulation » de cette extension de la théorie. Il s'agit là de travaux préliminaires qui réclameront des développements complémentaires dans les suites de ce travail de thèse, notamment pour raffiner la fonction de d-STDP. De plus, les mesures expérimentales montrent que la causalité entre les spikes pré- et post-synaptiques doit être répétée pour induire la plasticité [Bi and Poo, 1998] [Feldman, 2012]. Nous pourrions donc, par exemple, proposer un mode de mise à jour des poids qui ne soit pas instantané.

Pour conclure, ce chapitre met en exergue la souplesse de la modélisation des neurones proposée dans cette thèse. Comme démontré ici, il est possible de concevoir aisément des extensions pour s'affranchir d'hypothèses initialement posées.

CONCLUSION ET PERSPECTIVES

Conclusion

La théorie des assemblées de neurones et les nombreuses données expérimentales obtenues dans ce domaine constituent un cadre fondamental pour la compréhension du fonctionnement neuronal et cérébral (Chapitre 1). Cependant, leur étude *in vivo* et en particulier l'étude de leur mise en place structurelle et fonctionnelle est limitée par les contraintes expérimentales, qu'elles soient de nature technique (mesure de l'activité de plusieurs neurones, *etc.*), méthodologique (reproductibilité, interprétation des mesures, *etc.*) ou éthique (expérimentation sur animaux éveillés, *etc.*) Dans ce contexte, les approches *in silico* constituent un moyen d'investigation complémentaire indispensable pour tester des hypothèses, orienter les recherches, aider à la mise en place de méthodologies, *etc.* De plus, les modèles (à visée explicative) permettent d'intégrer des résultats d'expériences obtenus avec des techniques ou dans des contextes différents et ainsi de tester de façon intégrative la cohérence d'un plus grand nombre d'hypothèses que ce qui est envisageable expérimentalement.

Le sujet original de cette thèse a été proposé en partant du constat suivant : une assemblée de neurones est caractérisée par la synchronisation temporelle des activations des neurones qui la composent ; puisque les dendrites conditionnent l'intégration des potentiels post-synaptiques, elles agissent inévitablement sur l'atteinte du seuil de dépolarisation et les temps d'occurrence des spikes ; par conséquent, les modèles de neurones ponctuels seraient *a priori* fondamentalement incompatibles avec l'étude *in silico* des assemblées : il faudrait concevoir un modèle avec une représentation explicite des dendrites et de leur rôle.

Par ailleurs, cette thèse s'inscrit dans une démarche formelle, avec l'objectif à long terme de permettre du raisonnement assisté par ordinateur sur les neurones en réseaux. Ce type d'approche requiert donc de manipuler des modèles relativement abstraits. Or, les modèles satisfaisants de ce point de vue sont généralement ponctuels ou trop restrictifs (Chapitre 3). Face aux limites des modèles existants, notre rôle d'informaticiens est alors de les combiner de façon innovante, voire d'en inventer de nouveaux, pour satisfaire les besoins.

Un nouveau modèle de neurone.

Tout au long de cette thèse, nous avons développé un modèle de neurone compatible avec l'étude « explicative » *in silico* des assemblées : il a la particularité d'intégrer des dendrites

CONCLUSION ET PERSPECTIVES

sous une forme abstraite, tout en prenant en compte le rôle de l'arborescence dendritique dans l'intégration des signaux (Chapitre 4). Comme nous avons pu l'observer dans la Section 8.3 du Chapitre 8, la prise en compte des dendrites est déterminante lorsque l'on s'intéresse à la fonction d'entrée/sortie des neurones.

Notre modèle de neurone est un système qui convertit une multitude de signaux d'entrée discrets en signaux continus, qui sont ensuite intégrés puis reconvertis en un unique signal de sortie discret. Les signaux d'entrée, qui représentent des suites de spikes, sont reçus et convertis par les synapses. Puis, les signaux continus traversent l'arborescence dendritique, dont la forme n'est pas contrainte. L'arborescence est découpée en compartiments respectivement caractérisés par uniquement deux paramètres : le temps de traversée propre au compartiment et l'atténuation qu'il provoque sur le signal. De notre point de vue, il s'agit donc d'un modèle hybride, discret aux entrées/sorties, mais continu au niveau des états internes avec une infinité de points de l'arborescence possédant chacun un état dans \mathbb{R} .

Nous avons fait le choix original de considérer le signal en termes de charges et non de voltage. Ceci permet de traiter des sommes aux embranchements de l'arborescence, une opération qui s'avère intéressante pour la mise en évidence des propriétés remarquables du modèle. Le soma du neurone est modélisé à la manière d'un modèle classique de Leaky Integrate-and-Fire. Il intègre tous les signaux provenant des dendrites et le signal résultant subit une atténuation continue. Lorsque ce signal atteint un certain seuil, le neurone émet un spike sur sa sortie. Une originalité par rapport au modèle Leaky Integrate-and-Fire est que nous autorisons un seuil variable au cours du temps (fonction de borne), bien plus crédible biologiquement. Au moment d'atteindre ce seuil, la valeur du signal au soma est mise à jour et le neurone entre en période réfractaire, durant laquelle le seuil est inatteignable puis retourne progressivement vers sa valeur normale. Là encore, une originalité de notre modèle est que l'état du soma n'est pas remis à zéro, pour plus de crédibilité biologique.

Ce modèle repose sur des hypothèses choisies pour préserver des propriétés algébriques, tout en conservant une plausibilité biologique dans le contexte des assemblées de neurones (Chapitres 4 et 5). Nous avons montré, dans le Chapitre 9 que le formalisme que nous proposons peut être facilement enrichi, levant des hypothèses initialement posées. En particulier, nous avons introduit de la plasticité synaptique, un processus au cœur de la mise en place des assemblées et également majeur pour des applications d'apprentissage automatique.

Une définition rigoureuse et des propriétés remarquables.

La démarche formelle dans laquelle nous nous plaçons impose une définition mathématique du modèle que nous proposons. Nous avons donc établi une description rigoureuse de chaque élément du neurone et de sa dynamique. En particulier, nous avons fondé l'approche sur une description entièrement symbolique de chacun des éléments d'un neurone, ce qui est essentiel pour des applications formelles futures.

À partir de cette définition formelle de notre modèle, nous avons mis en évidence des propriétés remarquables. Nous avons prouvé qu'il est possible de réduire la structure des neurones

en préservant leur fonction d'entrée/sortie, c'est-à-dire en préservant la relation entre les spikes reçus et les spikes émis par les neurones (Chapitre 5). Nous avons ainsi identifié, pour tout neurone, une forme réduite dans laquelle chaque synapse est liée au soma du neurone par un compartiment au maximum. De plus, nous avons identifié une forme normalisée dans laquelle les compartiments ne portent aucune atténuation et le seuil au soma est normalisé à 1, hors périodes réfractaires. La forme normalisée représente canoniquement toute une classe d'équivalence de structures partageant la même fonction d'entrée/sortie. Cependant, la forme normalisée ne représente pas la classe d'équivalence de manière unique : il peut exister deux neurones normalisés différents ayant la même fonction d'entrée/sortie. Les hypothèses sous lesquelles les neurones normalisés sont effectivement des représentants uniques semblent pour le moins extrêmement *ad hoc* et nous n'avons trouvé aucune caractérisation rigoureuse (mais beaucoup de contre-exemples). Cette question reste donc ouverte.

En nous appuyant sur la théorie des catégories, nous avons démontré et étendu ces propriétés d'un point de vue purement algébrique (Chapitre 6). Nous avons mis en évidence, par une notion de morphisme, tout un continuum de neurones partageant la même fonction d'entrée/sortie. Les objets minimaux de la catégorie des neurones correspondent aux formes normalisées précédemment décrites. Nous avons également mis en lumière les structures algébriques sous-jacentes aux différents éléments du cadre de modélisation : les signaux d'entrée (discrets et leur conversion continue) ont une structure de monoïde, les états dendritiques ont une structure de groupe abélien, *etc.* Pour l'état du soma, une vision analytique s'impose à cause de l'équation différentielle impliquée, qui « casse » la continuité des propriétés algébriques de l'arborescence dendritique. Néanmoins, nous apportons dans cette thèse un premier pas vers une formalisation complète des neurones (sous forme d'un ensemble d'axiomes), avec de belles propriétés sur l'arborescence dendritique et un premier découpage en classes d'équivalence de l'ensemble des neurones.

Dans la perspective d'application des méthodes formelles, nous avons implémenté notre modèle de neurone en Lustre, avec une discrétisation temporelle préalable (Chapitres 7 et 8). Lustre est un langage formel qui bénéficie de model checkers vérifiant des propriétés directement encodées dans ce langage. Nous avons exprimé en Lustre la propriété d'équivalence entre un neurone et sa forme normalisée, avec des entrées constantes. Sur un exemple de neurone très simple, le model checker kind2 vérifie efficacement la satisfaction de cette propriété. Malheureusement, on atteint rapidement les limites du model checking en complexifiant les entrées. Néanmoins, nous avons montré qu'il est possible de prouver des propriétés de manière automatique, ce qui est encourageant. L'implémentation en Lustre, par essence, est également un indicateur sur la justesse de nos définitions formelles.

Perspectives

Vers les neurosciences *in silico*.

Les résultats présentés dans cette thèse ouvrent de nombreuses perspectives en neuro-

CONCLUSION ET PERSPECTIVES

ciences, à la fois à l'échelle d'un neurone unique et à l'échelle des réseaux.

À l'échelle du neurone, la direction de recherche privilégiée concerne le rôle des dendrites dans la fonction neuronale. Cette piste est motivée par les théorèmes de réduction et de normalisation que nous avons prouvés (Chapitre 5). Effectivement, ces théorèmes mettent en évidence des classes d'équivalence et nuancent le rôle de la structure précise des dendrites dans la fonction des neurones : l'arborisation dendritique ne contribuerait à la fonction d'entrée/sortie des neurones, que par l'apport de délais entre les synapses et le soma, dans un cadre de conduction passive. Notre point de vue est que l'atténuation est une conséquence de l'arborisation dendritique, qui pourrait être compensée par différents mécanismes (augmentation du poids des synapses, mécanismes actifs, *etc.*) contrairement aux délais qui sont incompressibles.

La mise en évidence des classes d'équivalence mène à une question intéressante : si la morphologie des neurones n'est pas dictée de manière unique par la fonction d'entrée/sortie, comment expliquer les morphologies typiques des neurones biologiques (Figure 1.2, Chapitre 1) ? Pour investiguer cette problématique, nous pourrions, à partir des formes normalisées, reconstruire les arborescences dendritiques qui préservent la fonction d'entrée/sortie sous contraintes. Par exemple, nous pourrions contraindre la reconstruction par des critères tels que la limitation des redondances, l'économie de membrane, l'optimisation de la surface de contact, *etc.* [Stuart et al., 2016]. L'idée serait alors d'identifier des critères menant aux formes caractéristiques des neurones biologiques. Ceci permettrait potentiellement d'expliquer pourquoi une morphologie est privilégiée vis-à-vis d'autres morphologies permettant de remplir la même fonction d'entrée/sortie.

Dans une direction similaire, nous pourrions développer une méthode de classification originale des neurones sur la base de leur fonction d'entrée/sortie, plutôt que sur celle de leur morphologie. On pourrait modéliser canoniquement les neurones biologiques sous forme normalisée. L'enjeu informatique serait alors de définir une notion de distance entre les neurones normalisés, permettant de les comparer et de les classer.

Il est essentiel de rappeler que notre modèle repose sur des hypothèses biologiques, permettant de préserver des propriétés algébriques intéressantes (Chapitres 4, 5 et 6). Nous pouvons donc considérer ce modèle comme une représentation algébriquement idéale, de ce qu'est un neurone. Finalement, notre formalisme pourrait orienter les recherches en neurosciences sur des problématiques traitant des *différences* avec nos neurones idéalisés, afin d'identifier d'autres propriétés fondamentales des neurones biologiques.

Pour être vraiment à la pointe des dernières avancées en neurosciences, il serait intéressant d'envisager l'introduction de propriétés actives dans les dendrites. Ce sont des propriétés encore peu et mal connues, et constituent une nouvelle frontière pour les recherches en biologie. De notre point de vue informatique, l'enjeu serait de prendre en compte de telles propriétés tout en conservant des propriétés algébriques remarquables.

En connectant les neurones en réseaux, on peut envisager d'étudier les mécanismes de mise

en place et de fonctionnement des assemblées de neurones. C'est cette direction qui a initialement motivé le cadre de modélisation proposé dans cette thèse. En particulier, on pourrait se focaliser sur le rôle des délais apportés par les dendrites. Dans ce contexte, la prise en compte de cette dimension temporelle est une particularité précieuse de notre modèle. Plus généralement, notre modèle pourrait permettre de mieux comprendre les phénomènes impliquant de la synchronisation ou de la coordination entre les neurones. On pourrait notamment s'intéresser à l'impact des délais dans le fonctionnement des microcircuits.

Par ailleurs, les dendrites sont impliquées dans de nombreuses pathologies [Stuart et al., 2016]. On peut notamment citer l'*épilepsie*. Cette pathologie, dont les causes sont multiples, se manifeste typiquement par des crises de convulsions répétées et se caractérise par une activité électrique anormale (hyperactivité) du cerveau. Ce dysfonctionnement proviendrait d'un déséquilibre des inhibitions neuronales. Plus précisément, l'inhibition dendritique des cellules principales de l'hippocampe serait fortement diminuée alors que l'inhibition somatique serait augmentée chez les patients épileptiques [Cossart et al., 2001]. Grâce à notre modèle, à partir des circuits impliqués connus, nous pourrions essayer de mieux comprendre l'influence du placement des synapses inhibitrices sur les activités épileptiques [Wendling et al., 2002]. Par exemple, l'activité anormale est-elle simplement due à une diminution globale de l'inhibition des cellules principales ou provient-elle particulièrement du dysfonctionnement des synapses distales ? Les délais permettent-ils d'expliquer les comportements caractéristiques observés (tels que l'arrêt complet des spikes en fin de crise [Truccolo et al., 2011]) ? Des réponses à ce type de questions pourraient permettre de mieux comprendre la fonction des dendrites et également dans des contextes pathologiques.

Ces recherches *in silico* pourraient se faire dans un premier temps par simulation, avec l'espoir à terme, d'obtenir des résultats par raisonnement automatique.

Vers les méthodes formelles.

Un intérêt des méthodes formelles est leur capacité de démonstration assistée par ordinateur. Il existe différentes techniques parmi lesquelles nous avons choisi le model checking. Nous avons réussi à prouver des propriétés simples, mais nous nous sommes très rapidement heurtés à des limitations. Le problème majeur du model checking est celui de l'explosion des états. La solution classique à ce problème est d'analyser une abstraction du système qui préserve ses propriétés. Dans notre cas, c'est probablement la puissance d'abstraction de kind2 qui est insuffisante. Cependant, kind2 est *a priori* le model checker le plus puissant pour Lustre [De Maria et al., 2016]. Nous pourrions donc envisager d'autres model checkers, mais ceci nécessiterait l'implémentation de notre modèle dans un format plus adapté, sans passer par l'intermédiaire de Lustre. Par ailleurs, nous pourrions également simplifier notre modèle, mais ceci pourrait remettre en cause sa plausibilité biologique. Nous avons choisi le model checking, car c'est une approche qui a fait ses preuves pour l'étude des systèmes biologiques, qui a l'avantage d'être automatique et de requérir fort peu d'expertise, mais nous pourrions également envisager d'autres techniques formelles.

La piste privilégiée dans cette direction, serait de formaliser complètement notre modèle de

CONCLUSION ET PERSPECTIVES

neurone, en vue d'établir une « algèbre de neurones ». Notre principal obstacle est de combiner la spécification algébrique de l'arborescence dendritique avec la spécification analytique du soma.

À long terme, nous aimerions identifier automatiquement les paramètres statiques des neurones pour des fonctions d'entrée/sortie particulières.

Vers l'informatique bio-inspirée.

Nous avons montré dans cette thèse que des délais permettent de représenter une intégration dendritique abstraite. C'est un moyen simple de prendre en compte ce phénomène, sans représenter explicitement l'arborescence dendritique. Dans de nombreux modèles, des délais sont déjà pris en compte, généralement pour représenter la transmission axonale et/ou synaptique. L'intégration dendritique est ignorée pour deux raisons principales : pour sa complexité, et, souvent pour la croyance que la durée de ce processus est négligeable devant la transmission axonale. Cependant, ces durées sont en fait du même ordre [Agmon-Snir and Segev, 1993] [Stuart et al., 2016]. Nous levons ici la limite de la complexité en montrant qu'il est pertinent de prendre en compte l'intégration dendritique avec de simples délais. Afin de limiter le nombre de paramètres impliqués, il serait même possible de combiner les délais de transmissions axonale, synaptique et dendritique dans un unique paramètre lorsque les neurones sont mis en réseaux.

Les réseaux de neurones sont utilisés pour les neurosciences, mais également, et très largement, pour l'apprentissage automatique. Dans ce contexte, ce sont généralement des neurones abstraits, proches du modèle de McCulloch et Pitts, qui sont utilisés (Chapitre 3).

Dans l'espoir de meilleures performances, des recherches visent à enrichir les réseaux d'apprentissage en plausibilité biologique. Dans les réseaux de neurones artificiels classiques, il n'y a pas vraiment de dimension temporelle. En particulier, les neurones n'accumulent pas d'information au cours du temps comme le ferait un neurone biologique. Pour remédier à cela, il est possible de faire apprendre des réseaux de neurones spikants (dits de *troisième génération*). Cependant, en pratique, les réseaux spikants n'atteignent pas les impressionnantes performances des réseaux de neurones classiques. Ceci s'explique notamment par le manque de techniques d'apprentissage supervisé pour les réseaux de neurones spikants [Mostafa, 2018]. L'algorithme de *rétro-propagation*, le plus utilisé et performant, ne peut être appliqué aux spikes car ils ne sont pas dérivables. Les réseaux de neurones spikants n'en restent pas moins intéressants. Il a été montré, qu'en théorie et au regard du nombre de neurones requis, les réseaux de neurones spikants sont plus puissants que les réseaux de neurones de première et deuxième générations [Maass, 1997]. De plus, ils sont plus compatibles avec l'implémentation matérielle et moins demandeurs en énergie [Tavanaei et al., 2018].

L'apprentissage par les réseaux de neurones spikants se fonde préférentiellement sur la STDP [Masquelier and Thorpe, 2007] [Masquelier et al., 2008]. Dans ce contexte, l'introduction de la plasticité à notre modèle ouvre des perspectives intéressantes. D'un point de vue biologique, nous avons vu qu'il serait plus pertinent de considérer une loi STDP par

connexion, pour tenir compte de l'intégration dendritique. On peut alors se poser la question suivante : cette adaptation de la STDP bénéficierait-elle à l'apprentissage automatique ?

Comme on le voit, nos travaux ouvrent de nombreuses perspectives et la plupart d'entre elles n'étaient pas attendues à l'origine. Ces perspectives sont nombreuses et nous espérons que notre formalisme, à la croisée de la biologie théorique et de l'informatique fondamentale, facilitera de nouvelles recherches dans ces deux domaines.

ANNEXES

Annexe A Code Lustre - partie 1

Nous fournissons ici le code Lustre nécessaire pour prouver par model checking l'équivalence de la sortie des forêts dendritiques des neurones illustrés en Figure 8.6 (Chapitre 8), pour des entrées constantes.

Nœud 9: Synapse inhibitrice (I) de paramètres $w = -5$, $\hat{\tau} = 5$, $\tilde{\tau} = 10$.

```
node synapseI5_5_10 (I_s : bool) returns (eta_s : real);
  var w, hatTau, checkTau : real;
      o1, o2, o3, o4, o5, o6, o7, o8, o9, o10, o11, o12,
      o13, o14 : bool;
  let
    w = -5.0; hatTau = 5.0; checkTau = 10.0;
    o1 = false -> pre(I_s);
    o2 = false -> pre(o1);
    o3 = false -> pre(o2);
    o4 = false -> pre(o3);
    o5 = false -> pre(o4);
    o6 = false -> pre(o5);
    o7 = false -> pre(o6);
    o8 = false -> pre(o7);
    o9 = false -> pre(o8);
    o10 = false -> pre(o9);
    o11 = false -> pre(o10);
    o12 = false -> pre(o11);
    o13 = false -> pre(o12);
    o14 = false -> pre(o13);
    eta_s = 0.0 -> (if o1 then w/hatTau else 0.0)
                  + (if o2 then 2.0*w/hatTau else 0.0)
                  + (if o3 then 3.0*w/hatTau else 0.0)
                  + (if o4 then 4.0*w/hatTau else 0.0)
                  + (if o5 then 5.0*w/hatTau else 0.0)
                  + (if o6 then w-w/checkTau else 0.0)
                  + (if o7 then w-2.0*w/checkTau else 0.0)
                  + (if o8 then w-3.0*w/checkTau else 0.0)
                  + (if o9 then w-4.0*w/checkTau else 0.0)
                  + (if o10 then w-5.0*w/checkTau else 0.0)
                  + (if o11 then w-6.0*w/checkTau else 0.0)
                  + (if o12 then w-7.0*w/checkTau else 0.0)
                  + (if o13 then w-8.0*w/checkTau else 0.0)
                  + (if o14 then w-9.0*w/checkTau else 0.0);
  tel
```

Nœud 10: Synapse excitatrice (E) de parametres $w = 5$, $\hat{\tau} = 5$, $\check{\tau} = 15$.

```

node synapseE5_5_15 (I_s : bool) returns (eta_s : real);
  var w, hatTau, checkTau : real;
    o1, o2, o3, o4, o5, o6, o7, o8, o9, o10, o11, o12,
    o13, o14, o15, o16, o17, o18, o19 : bool;
  let
    w = 5.0; hatTau = 5.0; checkTau = 15.0;
    o1 = false -> pre(I_s);
    o2 = false -> pre(o1);
    o3 = false -> pre(o2);
    o4 = false -> pre(o3);
    o5 = false -> pre(o4);
    o6 = false -> pre(o5);
    o7 = false -> pre(o6);
    o8 = false -> pre(o7);
    o9 = false -> pre(o8);
    o10 = false -> pre(o9);
    o11 = false -> pre(o10);
    o12 = false -> pre(o11);
    o13 = false -> pre(o12);
    o14 = false -> pre(o13);
    o15 = false -> pre(o14);
    o16 = false -> pre(o15);
    o17 = false -> pre(o16);
    o18 = false -> pre(o17);
    o19 = false -> pre(o18);
    eta_s = 0.0 -> ( if o1 then w/hatTau else 0.0)
                  + (if o2 then 2.0*w/hatTau else 0.0)
                  + (if o3 then 3.0*w/hatTau else 0.0)
                  + (if o4 then 4.0*w/hatTau else 0.0)
                  + (if o5 then 5.0*w/hatTau else 0.0)
                  + (if o6 then w-w/checkTau else 0.0)
                  + (if o7 then w-2.0*w/checkTau else 0.0)
                  + (if o8 then w-3.0*w/checkTau else 0.0)
                  + (if o9 then w-4.0*w/checkTau else 0.0)
                  + (if o10 then w-5.0*w/checkTau else 0.0)
                  + (if o11 then w-6.0*w/checkTau else 0.0)
                  + (if o12 then w-7.0*w/checkTau else 0.0)
                  + (if o13 then w-8.0*w/checkTau else 0.0)
                  + (if o14 then w-9.0*w/checkTau else 0.0)
                  + (if o15 then w-10.0*w/checkTau else 0.0)
                  + (if o16 then w-11.0*w/checkTau else 0.0)
                  + (if o17 then w-12.0*w/checkTau else 0.0)
                  + (if o18 then w-13.0*w/checkTau else 0.0)
                  + (if o19 then w-14.0*w/checkTau else 0.0);
  tel

```

Nœud 11: Synapse excitatrice (E) de paramètres $w = 3$, $\hat{\tau} = 6$, $\check{\tau} = 12$.

```
node synapseE3_6_12 (I_s : bool) returns (eta_s : real);
  var w, hatTau, checkTau : real;
      o1, o2, o3, o4, o5, o6, o7, o8, o9, o10, o11, o12,
      o13, o14, o15, o16, o17 : bool;
  let
    w = 3.0; hatTau = 6.0; checkTau = 12.0;
    o1 = false -> pre(I_s);
    o2 = false -> pre(o1);
    o3 = false -> pre(o2);
    o4 = false -> pre(o3);
    o5 = false -> pre(o4);
    o6 = false -> pre(o5);
    o7 = false -> pre(o6);
    o8 = false -> pre(o7);
    o9 = false -> pre(o8);
    o10 = false -> pre(o9);
    o11 = false -> pre(o10);
    o12 = false -> pre(o11);
    o13 = false -> pre(o12);
    o14 = false -> pre(o13);
    o15 = false -> pre(o14);
    o16 = false -> pre(o15);
    o17 = false -> pre(o16);
    eta_s = 0.0 -> (if o1 then w/hatTau else 0.0)
                  + (if o2 then 2.0*w/hatTau else 0.0)
                  + (if o3 then 3.0*w/hatTau else 0.0)
                  + (if o4 then 4.0*w/hatTau else 0.0)
                  + (if o5 then 5.0*w/hatTau else 0.0)
                  + (if o6 then 6.0*w/hatTau else 0.0)
                  + (if o7 then w-w/checkTau else 0.0)
                  + (if o8 then w-2.0*w/checkTau else 0.0)
                  + (if o9 then w-3.0*w/checkTau else 0.0)
                  + (if o10 then w-4.0*w/checkTau else 0.0)
                  + (if o11 then w-5.0*w/checkTau else 0.0)
                  + (if o12 then w-6.0*w/checkTau else 0.0)
                  + (if o13 then w-7.0*w/checkTau else 0.0)
                  + (if o14 then w-8.0*w/checkTau else 0.0)
                  + (if o15 then w-9.0*w/checkTau else 0.0)
                  + (if o16 then w-10.0*w/checkTau else 0.0)
                  + (if o17 then w-11.0*w/checkTau else 0.0);
  tel
```

Nœud 12: Compartiment de parametres $\Delta = 1\delta$ et $\alpha = 0.5$.

```
node comp1 (I : real) returns (0 : real);
  var alphaSubC, init0 : real;
  let
    alphaSubC = 0.5;
    init0 = 0.0;
    0 = init0 -> pre(I) * alphaSubC;
  tel
```

Nœud 13: Compartiment de parametres $\Delta = 2\delta$ et $\alpha = 0.25$.

```
node comp2 (I : real) returns (0 : real);
  var alphaSubC, init0,
      c1, init1 : real;
  let
    alphaSubC = 0.5;
    init0 = 0.0;
    init1 = 0.0;
    c1 = init1 -> pre(I) * alphaSubC;
    0 = init0 -> pre(c1) * alphaSubC;
  tel
```

Nœud 14: Compartiment de parametres $\Delta = 3\delta$ et $\alpha = 0.125$.

```
node comp3 (I : real) returns (0 : real);
  var alphaSubC, init0,
      c1, c2, init1, init2 : real;
  let
    alphaSubC = 0.5;
    init0 = 0.0;
    init1 = 0.0; init2 = 0.0;
    c1 = init1 -> pre(I) * alphaSubC;
    c2 = init2 -> pre(c1) * alphaSubC;
    0 = init0 -> pre(c2) * alphaSubC;
  tel
```

ANNEXES

Nœud 15: Neurone1 (uniquement foret dendritique, sans soma).

```
node neurone1 (I1, I2, I3 : bool) returns (F : real);
  var S1, S2, S3, C1, C2, C3, C23 : real;
  let
    S1 = synapseE5_5_15 (I1);
    S2 = synapseE3_6_12 (I2);
    S3 = synapseI5_5_10 (I3);
    C1 = comp1 (S1);
    C2 = comp1 (S2);
    C3 = comp1 (S3);
    C23 = comp1 (C2 + C3);
    F = comp1 (C1 + C23);
  tel
```

Nœud 16: Neurone2 (uniquement foret dendritique, sans soma).

```
node neurone2 (I1, I2, I3 : bool) returns (F : real);
  var S1, S2, S3, C1, C2, C3,
      initF : real;
  let
    initF = 0.0;
    S1 = synapseE5_5_15 (I1);
    S2 = synapseE3_6_12 (I2);
    S3 = synapseI5_5_10 (I3);
    C1 = comp2 (S1);
    C2 = comp3 (S2);
    C3 = comp3 (S3);
    F = initF -> C1 + C2 + C3;
  tel
```

Nœud 17: Propriete d'equivalence avec signaux d'entree constants.

```
node equivalence (gost : bool) returns (prop: bool);
  var I1, I2, I3 : bool;
  let
    I1 = true;
    I2 = true;
    I3 = false;
    prop = neurone1 (I1, I2, I3) = neurone2 (I1, I2, I3);
    --@ --%MAIN;
    --@ --%PROPERTY prop;
  tel
```

Annexe B Code Lustre - partie 2

Nous fournissons ici le code Lustre nécessaire pour prouver par model checking l'équivalence de la sortie des neurones illustrés en Figure 8.6, pour des entrées constantes. La différence avec le code présenté en Annexe A provient de la prise en compte du soma qui n'est pas nécessaire pour prouver l'équivalence. Les paramètres du soma sont choisis de manière à ce que le neurone soit capable d'émettre des spikes sur sa sortie. Les appels à `neurone1` et `neurone2` renvoient respectivement aux nœuds `n1` et `n2` de l'Annexe A.

Nœud 18: Soma de parametres $p_\beta = 10$, $\hat{p}_\beta = 10$, $e_\beta = 10$, $\hat{e}_\beta = 50$ et $\gamma = 0.1$.

```
node soma (F : real) returns (O : bool);
  var eBeta, hat_eBeta, pBeta, hat_pBeta, gamma,
      varpBeta, dp, e, p, e0, p0 : real;
      PRA, PRR, normal : bool;
  let
    pBeta = 10.0; hat_pBeta = 10.0;
    eBeta = 10.0; hat_eBeta = 50.0;
    gamma = 0.1;
    e0 = 1000.0; p0 = 0.0;
    PRA = false -> if (pre(e) + 1 > eBeta) then false else true;
    PRR = false -> if (pre(e) + 1 > eBeta and pre(e) + 1 <= eBeta + hat_eBeta)
                    then true
                    else false;
    normal = true -> if (PRA or PRR) then false else true;
    varpBeta = pBeta + hat_pBeta -> if PRR
                                    then pre(varpBeta) - (hat_pBeta/hat_eBeta)
                                    else pBeta + hat_pBeta;

    dp = 0.0 -> F - gamma * pre(p);
    p = p0 -> if ((normal and pre(p) + dp < pBeta)
                  or (PRR and pre(p) + dp < varpBeta)
                  or (PRA))
              then pre(p) + dp
              else pre(p) + dp - pBeta;
    O = false -> if ((normal and pre(p) + dp >= pBeta)
                    or (PRR and pre(p) + dp >= varpBeta))
                then true
                else false;
    e = e0 -> if (O) then 0.0 else pre(e) + 1.0;
  tel
```

ANNEXES

Nœud 19: Neurone1 (avec soma).

```
node neurone1Soma (I1, I2, I3 : bool) returns (O : bool);
  var F : real;
  let
    F = neurone1 (I1, I2, I3);
    O = soma (F);
  tel
```

Nœud 20: Neurone2 (avec soma).

```
node neurone2Soma (I1, I2, I3 : bool) returns (O : bool);
  var F : real;
  let
    F = neurone2 (I1, I2, I3);
    O = soma (F);
  tel
```

Nœud 21: Propriete d'equivalence entre Neurone1 et Neurone2 (avec soma), pour des entrees constantes.

```
node equivalenceSoma (gost : bool) returns (prop: bool);
  var I1, I2, I3 : bool;
  let
    I1 = false;
    I2 = true;
    I3 = false;
    prop = neurone1Soma (I1, I2, I3) = neurone2Soma (I1, I2, I3);
    --%MAIN;
    --%PROPERTY prop;
  tel
```

Annexe C Code Lustre - partie 3

Nous fournissons ici le code Lustre nécessaire pour prouver par model checking l'équivalence de la sortie des neurones illustrés en Figure 8.6, pour des entrées régulières (non constantes). Les variables locales `cpt1`, `cpt2`, `cpt3` servent de compteurs de pas de temps, permettant de définir la fréquence des signaux d'entrée.

Nœud 22: Propriété d'équivalence avec signaux d'entrée réguliers.

```
node equivalence_Frq (gost : bool) returns (prop: bool);
  var I1, I2, I3 : bool;
      cpt1, cpt2, cpt3 : int;
  let
    cpt1 = 1 -> if pre(cpt1) < 50 then pre(cpt1) + 1 else 1;
    cpt2 = 1 -> if pre(cpt2) < 10 then pre(cpt2) + 1 else 1;
    cpt3 = 1 -> if pre(cpt3) < 50 then pre(cpt3) + 1 else 1;
    I3 = (cpt1 = 50); --@ I1 true 1 instant sur 50
    I1 = (cpt2 = 10);
    I2 = (cpt3 = 20);
    prop = neurone1 (I1, I2, I3) = neurone2 (I1, I2, I3);
    --%MAIN;
    --%PROPERTY prop;
  tel
```

Annexe D Code Java permettant de générer du code Lustre

Nous fournissons ici la classe permettant de générer le code Lustre implémentant une synapse. Lors de l'instanciation d'une synapse, il suffit de fournir ses paramètres w , $\hat{\tau}$ et $\check{\tau}$ ainsi qu'un nom qui sera celui du nœud correspondant. La méthode `writeLus` permet d'écrire le nœud dans un fichier (`.lus`) dont le nom est pris en paramètre.

```
import java.util.ArrayList;
import java.io.*;

public class Synapse{

    // variables d'instance
    private String nodeName;
    private int w;
    private int hatTau;
    private int checkTau;

    // constructeur
    public Synapse(String name, int weight, int timeToPeak, int
        descentDelay){
        this.nodeName = name;
        this.w = weight;
        this.hatTau = timeToPeak;
        this.checkTau = descentDelay;
    }

    //accesseur
    public String getNodeName(){
        return(this.nodeName);
    }

    // creation du noeud Lustre correspondant - etat initial = 0.0
    public String toString(){
        String node = "node " + this.nodeName + " (I_s : bool) returns (
            eta_s : real);\n";
        ArrayList<String> pastSpikes = new ArrayList<String>();
        for (int i = 1; i < this.hatTau + this.checkTau ; i++){
            String o = "o" + i;
            pastSpikes.add(o);
        }
        node += " var w, hatTau, checkTau : real; ";
        for(int i = 0; i < pastSpikes.size()-1; i++){
            node = node + pastSpikes.get(i) + ", ";
        }
        node = node + pastSpikes.get(pastSpikes.size()-1) + " : bool;\n";
        node += " let\n";
    }
}
```

D. CODE JAVA PERMETTANT DE GÉNÉRER DU CODE LUSTRE

```
node = node + " w = " + this.w + ".0;\n";
node = node + " hatTau = " + this.hatTau + ".0;\n";
node = node + " checkTau = " + this.checkTau + ".0;\n";
node = node + " " + pastSpikes.get(0) + " = false -> pre(I_s)
    ;\n";
for(int i = 1; i < pastSpikes.size(); i++){
    node = node + " " + pastSpikes.get(i) + " = false -> pre(" +
        pastSpikes.get(i-1) + ");\n";
}
node = node + " eta_s = 0.0 -> (if " + pastSpikes.get(0) + "
    then w/hatTau else 0.0) \n ";
for(int i = 1; i < this.hatTau; i++){
    node = node + "+ (if " + pastSpikes.get(i) + " then " + (i+1) +
        ".0*w/hatTau else 0.0) \n ";
}
for(int i = this.hatTau; i < pastSpikes.size()-1; i++){
    node = node + "+ (if " + pastSpikes.get(i) + " then w-" + (i-this
        .hatTau+1) + ".0*w/checkTau else 0.0) \n ";
}
node = node + "+ (if " + pastSpikes.get(pastSpikes.size()-1) + "
    then w-" + (this.checkTau-1) + ".0*w/checkTau else 0.0);\n";
node += " tel \n\n";
return(node);
}

// ecriture dans fichier
public void writeLus(String fileName) { /
    String node = this.toString();
    FileWriter lus = null;
    try{
        lus = new FileWriter(fileName,true);
        lus.write(node);
        lus.close();
    } catch (IOException exception) {
        exception.printStackTrace();
    }
}
}
```

Annexe E Complément Figure 8.8, Chapitre 8

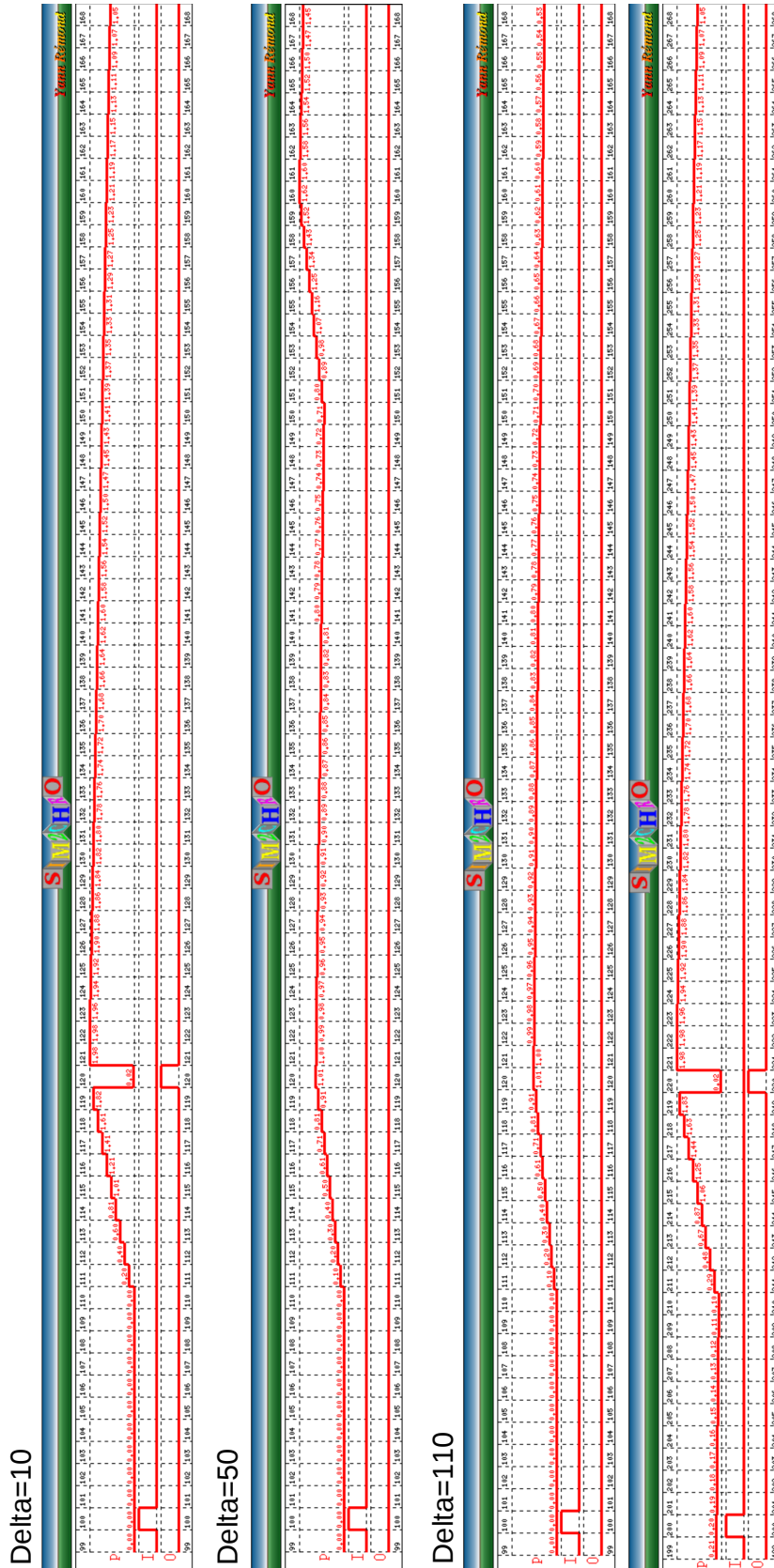


FIGURE 9.5 – Évolution, sur une fenêtre temporelle choisie, de la valeur du potentiel p au soma (première ligne) et du signal de sortie O (dernière ligne) en réponse au signal d'entrée I (deuxième ligne) identiquement reçu sur chaque synapse. Les différentes valeurs de Δ correspondent au temps de traversée du compartiment variable de la Figure 8.7 du Chapitre 8.

BIBLIOGRAPHIE

- [Abbott, 1999] Abbott, L. (1999). Lapique’s introduction of the integrate-and-fire model neuron (1907). *Brain research bulletin*, 50(5-6) :303–304.
- [Abeles, 2012] Abeles, M. (2012). *Local cortical circuits : An electrophysiological study*, volume 6. Springer Science & Business Media.
- [Adrian, 1928] Adrian, E. (1928). *The basis of sensation*. Christophers ; London, 22 Berners Steet, W. 1.
- [Agmon-Snir and Segev, 1993] Agmon-Snir, H. and Segev, I. (1993). Signal delay and input synchronization in passive dendritic structures. *Journal of neurophysiology*, 70(5) :2066–2085.
- [Ahmed et al., 2017] Ahmed, O., Kakarash, Z., and Ahmed, A. (2017). The dimensions of petri nets : modelling strategies for biological aspects application to e.coli. *Journal of Theoretical & Applied Information Technology*, 95(19).
- [Aihara et al., 1990] Aihara, K., Takabe, T., and Toyoda, M. (1990). Chaotic neural networks. *Physics letters A*, 144(6-7) :333–340.
- [Alberts et al., 2002] Alberts, B., Johnson, A., Lewis, J., Walter, P., Raff, M., and Roberts, K. (2002). *Molecular biology of the cell 4th edition : International student edition*.
- [Almeida-Filho et al., 2014] Almeida-Filho, D., Lopes-dos Santos, V., Vasconcelos, N., Miranda, J., Tort, A., and Ribeiro, S. (2014). An investigation of hebbian phase sequences as assembly graphs. *Frontiers in neural circuits*, 8 :34.
- [Alur and Dill, 1994] Alur, R. and Dill, D. L. (1994). A theory of timed automata. *Theoretical computer science*, 126(2) :183–235.
- [Alvarez-Leefmans and Delpire, 2009] Alvarez-Leefmans, F. and Delpire, E. (2009). Thermodynamics and kinetics of chloride transport in neurons : an outline. *Physiology and pathology of chloride transporters and channels in the nervous system (Alvarez-Leefmans FJ, Delpire E, eds)*, pages 81–108.

BIBLIOGRAPHIE

- [André, 1996] André, C. (1996). Representation and analysis of reactive behaviors : A synchronous approach. In *Computational Engineering in Systems Applications (CESA)*, pages 19–29, Lille (F). IEEE-SMC.
- [André, 2005] André, C. (2005). Comparaison des styles de programmation de langages synchrones. *Projet AOSTE-Juin*, 5.
- [Augustine et al., 2004] Augustine, G., Fitzpatrick, D., Hall, W., LaMantia, A., Purves, D., White, L., et al. (2004). *Neurosciences : Third Edition*. Sinauer Associates.
- [Badel et al., 2008] Badel, L., Lefort, S., Berger, T., Petersen, C., Gerstner, W., and Richardson, M. (2008). Extracting non-linear integrate-and-fire models from experimental data using dynamic i–v curves. *Biological cybernetics*, 99(4-5) :361.
- [Baier and Katoen, 2008] Baier, C. and Katoen, J. (2008). Principles of model checking (representation and mind series).[sl].
- [Baladron et al., 2012] Baladron, J., Fasoli, D., Faugeras, O., and Touboul, J. (2012). Mean-field description and propagation of chaos in networks of hodgkin-huxley and fitzhugh-nagumo neurons. *The Journal of Mathematical Neuroscience*, 2(1) :10.
- [Barrio et al., 2017] Barrio, R., Ibáñez, S., and Pérez, L. (2017). Hindmarsh–rose model : Close and far to the singular limit. *Physics Letters A*, 381(6) :597–603.
- [Basbaum, 2008] Basbaum, A. (2008). *The Senses : A Comprehensive Reference*. Academic Press.
- [Baumann and Pham-Dinh, 2001] Baumann, N. and Pham-Dinh, D. (2001). Biology of oligodendrocyte and myelin in the mammalian central nervous system. *Physiological reviews*, 81(2) :871–927.
- [Behaegel et al., 2016] Behaegel, J., Comet, J., Bernot, G., Cornillon, E., and Delaunay, F. (2016). A hybrid model of cell cycle in mammals. *Journal of bioinformatics and computational biology*, 14(01) :1640001.
- [Behaegel et al., 2018] Behaegel, J., Comet, J., and Pelleau, M. (2018). Identification de parametres dynamiques de réseaux de genes. *JFPC 2018*, page 33.
- [Bengio et al., 2009] Bengio, Y. et al. (2009). Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1) :1–127.
- [Benveniste et al., 1991] Benveniste, A., Le Guernic, P., and Jacquemot, C. (1991). Synchronous programming with events and relations : the signal language and its semantics. *Science of computer programming*, 16(2) :103–149.
- [Bernot et al., 2008] Bernot, G., Comet, J., and Khalis, Z. (2008). Gene regulatory networks with multiplexes. In *European simulation and modelling conference proceedings*, pages 423–432.
- [Bernot et al., 2004] Bernot, G., Comet, J., Richard, A., and Guespin, J. (2004). Application of formal methods to biological regulatory networks : extending thomas’ asynchronous logical approach with temporal logic. *Journal of theoretical biology*, 229(3) :339–347.

- [Berry, 1989] Berry, G. (1989). *Real time programming : Special purpose or general purpose languages*. PhD thesis, INRIA.
- [Berry and Cosserat, 1984] Berry, G. and Cosserat, L. (1984). The esterel synchronous programming language and its mathematical semantics. In *International Conference on Concurrency*, pages 389–448. Springer.
- [Berry and Gonthier, 1992] Berry, G. and Gonthier, G. (1992). The esterel synchronous programming language : Design, semantics, implementation. *Science of computer programming*, 19(2) :87–152.
- [Bezanilla, 2006] Bezanilla, F. (2006). The action potential : from voltage-gated conductances to molecular structures. *Biological research*, 39(3) :425–435.
- [Bi and Poo, 1998] Bi, G. and Poo, M. (1998). Synaptic modifications in cultured hippocampal neurons : dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of neuroscience*, 18(24) :10464–10472.
- [Biedermann-Thorson et al., 2013] Biedermann-Thorson, M., Dudel, J., Jänig, W., Schmidt, R., Schmidt, R., and Zimmermann, M. (2013). *Fundamentals of Neurophysiology*. Springer Study Edition. Springer Berlin Heidelberg.
- [Biere et al., 2009] Biere, A., Heule, M., and van Maaren, H. (2009). *Handbook of satisfiability*, volume 185. IOS press.
- [Bishop, 1995] Bishop, C. (1995). *Neural networks for pattern recognition*. Oxford university press.
- [Bobot, 2011] Bobot, F. (2011). *Logique de séparation et vérification déductive*. PhD thesis, Paris 11.
- [Börger and Nectow, 2013] Börger, C. and Nectow, A. (2013). Exponential time differencing for hodgkin–huxley-like odes. *SIAM Journal on Scientific Computing*, 35(3) :B623–B643.
- [Bourcerie, 2010] Bourcerie, M. (2010). La modélisation des systèmes de production pas réseaux de pétri. [Page disponible en ligne en juillet 2018].
- [Bower, 1998] Bower, J. (1998). Constructing new models. In *The Book of GENESIS*, pages 195–201. Springer.
- [Bradley, 2012] Bradley, A. (2012). Understanding ic3. In *International Conference on Theory and Applications of Satisfiability Testing*, pages 1–14. Springer.
- [Brette, 2012] Brette, R. (2012). Computing with neural synchrony. *PLoS computational biology*, 8(6) :e1002561.
- [Brette and Gerstner, 2005] Brette, R. and Gerstner, W. (2005). Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. *Journal of neurophysiology*, 94(5) :3637–3642.
- [Brette and Goodman, 2008] Brette, R. and Goodman, D. (2008). Brian 1.4.4 documentation » getting started » tutorials » tutorial 1 : Basic concepts ». <http://www>.

BIBLIOGRAPHIE

- briansimulator.org/docs/tutorial_1a_the_simplest_brian_program.html. [Page disponible en ligne en juillet 2018].
- [Brian, 2007] Brian, S. (2007). Méthodes numériques de résolution d'équations différentielles. http://www.fresnel.fr/perso/stout/Anal_numer/Cours4.pdf. [Page disponible en ligne en juin 2018].
- [Brock et al., 1952] Brock, L., Coombs, J., and Eccles, J. (1952). The recording of potentials from motoneurons with an intracellular electrode. *The Journal of Physiology*, 117(4) :431–460.
- [Brunel and Van Rossum, 2007] Brunel, N. and Van Rossum, M. (2007). Lapicque's 1907 paper : from frogs to integrate-and-fire. *Biological cybernetics*, 97(5-6) :337–339.
- [Budd and Kisvárdy, 2012] Budd, J. and Kisvárdy, Z. (2012). Communication and wiring in the cortical connectome. *Frontiers in neuroanatomy*, 6 :42.
- [bulletin officiel, 2011] bulletin officiel (2011). Programme de l'enseignement spécifique. http://media.education.gouv.fr/file/special_8_men/01/2/SVT_S_197012.pdf. [Page disponible en ligne en août 2018].
- [Butts et al., 2007] Butts, D., Weng, C., Jin, J., Yeh, C., Lesica, N., Alonso, J., and Stanley, G. (2007). Temporal precision in the neural code and the timescales of natural vision. *Nature*, 449(7158) :92.
- [Buzsáki, 2004] Buzsáki, G. (2004). Large-scale recording of neuronal ensembles. *Nature neuroscience*, 7(5) :446.
- [Calzone et al., 2006a] Calzone, L., Chabrier-Rivier, N., Fages, F., and Soliman, S. (2006a). Machine learning biochemical networks from temporal logic properties. In *Transactions on Computational Systems Biology VI*, pages 68–94. Springer.
- [Calzone et al., 2006b] Calzone, L., Fages, F., and Soliman, S. (2006b). Biocham : an environment for modeling biological systems and formalizing experimental knowledge. *Bioinformatics*, 22(14) :1805–1807.
- [Castillo et al., 2011] Castillo, P., Chiu, C., and Carroll, R. (2011). Long-term plasticity at inhibitory synapses. *Current opinion in neurobiology*, 21(2) :328–338.
- [Catterall et al., 2012] Catterall, W., Raman, I., Robinson, H., Sejnowski, T., and Paulsen, O. (2012). The hodgkin-huxley heritage : from channels to circuits. *Journal of Neuroscience*, 32(41) :14064–14073.
- [Champion et al., 2016] Champion, A., Mebsout, A., Sticksel, C., and Tinelli, C. (2016). The kind 2 model checker. In *International Conference on Computer Aided Verification*, pages 510–517. Springer.
- [Chklovskii et al., 2002] Chklovskii, D., Schikorski, T., and Stevens, C. (2002). Wiring optimization in cortical circuits. *Neuron*, 34(3) :341–347.
- [Clarke, 2008] Clarke, E. M. (2008). The birth of model checking. In *25 Years of Model Checking*, pages 1–26. Springer.

-
- [Collet et al., 2001] Collet, C., Maraninchi, F., and Potet, M. (2001). Compilation efficace d'itérateurs de tableaux lustre.
- [Comet, 2014] Comet, J. (2014). Les réseaux de petri pour la simulation de systèmes biologiques. [Page disponible en ligne en juillet 2018].
- [Coombs et al., 1955] Coombs, J., Eccles, J., and Fatt, P. (1955). Excitatory synaptic action in motoneurons. *The Journal of physiology*, 130(2) :374–395.
- [Cornillon, 2017] Cornillon, E. (2017). *Modèles qualitatifs de réseaux génétiques : réduction de modèles et introduction d'un temps continu*. PhD thesis, Université Côte d'Azur.
- [Cossart et al., 2001] Cossart, R., Dinocourt, C., Hirsch, J., Merchan-Perez, A., De Felipe, J., Ben-Ari, Y., Esclapez, M., and Bernard, C. (2001). Dendritic but not somatic gabaergic inhibition is decreased in experimental epilepsy. *Nature neuroscience*, 4(1) :52.
- [Cousot, 2000] Cousot, P. (2000). Interprétation abstraite. *Technique et science informatique*, 19(1-2) :3.
- [Cousot and Cousot, 1977] Cousot, P. and Cousot, R. (1977). Abstract interpretation : a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Proceedings of the 4th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*, pages 238–252. ACM.
- [Cuntz et al., 2010] Cuntz, H., Forstner, F., Borst, A., and Häusser, M. (2010). One rule to grow them all : a general theory of neuronal branching and its practical application. *PLoS computational biology*, 6(8) :e1000877.
- [Cybenko, 1989] Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4) :303–314.
- [Datadien, 2010] Datadien, A. (2010). The right delay : Detecting spike patterns with stdp and axonal delays.
- [Davis and de No Lorente, 1947] Davis, J. and de No Lorente, R. (1947). Contribution to the mathematical theory of the electrotonus. *Studies from the Rockefeller institute for medical research. Reprints. Rockefeller Institute for Medical Research*, 131 :442–496.
- [De Carlos and Borrell, 2007] De Carlos, J. and Borrell, J. (2007). A historical reflection of the contributions of cajal and golgi to the foundations of neuroscience. *Brain research reviews*, 55(1) :8–16.
- [De Jong, 2002] De Jong, H. (2002). Modeling and simulation of genetic regulatory systems : a literature review. *Journal of computational biology*, 9(1) :67–103.
- [De Maria et al., 2017] De Maria, E., Di Giusto, C., and Ciatto, G. (2017). Formal validation of neural networks as timed automata. In *Proceedings of the 8th International Conference on Computational Systems-Biology and Bioinformatics*, pages 15–22. ACM.
- [De Maria et al., 2016] De Maria, E., Muzy, A., Gaffé, D., Ressouche, A., and Grammont, F. (2016). Verification of temporal properties of neuronal archetypes modeled as synchronous reactive systems. In *International Workshop on Hybrid Systems Biology*, pages 97–112. Springer.

BIBLIOGRAPHIE

- [De N3 and Condouris, 1959] De N3, R. and Condouris, G. (1959). Decremental conduction in peripheral nerve. integration of stimuli in the neuron. *Proceedings of the National Academy of Sciences*, 45(4) :592–617.
- [Delarue et al., 2015] Delarue, F., Inglis, J., Rubenthaler, S., and Tanr3, E. (2015). Particle systems with a singular mean-field self-excitation. application to neuronal networks. *Stochastic Processes and their Applications*, 125(6) :2451–2492.
- [Denil et al., 2013] Denil, M., Shakibi, B., Dinh, L., De Freitas, N., et al. (2013). Predicting parameters in deep learning. In *Advances in neural information processing systems*, pages 2148–2156.
- [Denk et al., 1990] Denk, W., Strickler, J., and Webb, W. (1990). Two-photon laser scanning fluorescence microscopy. *Science*, 248(4951) :73–76.
- [Diesmann et al., 1999] Diesmann, M., Gewaltig, M., and Aertsen, A. (1999). Stable propagation of synchronous spiking in cortical neural networks. *Nature*, 402(6761) :529.
- [Dijkstra, 1976] Dijkstra, E. (1976). *A discipline of programming*, volume 1. prentice-hall Englewood Cliffs.
- [Dodt and Zieglg3nsberger, 1990] Dodt, H. and Zieglg3nsberger, W. (1990). Visualizing unstained neurons in living brain slices by infrared dic-videomicroscopy. *Brain research*, 537(1-2) :333–336.
- [D’amour and Froemke, 2015] D’amour, J. and Froemke, R. (2015). Inhibitory and excitatory spike-timing-dependent plasticity in the auditory cortex. *Neuron*, 86(2) :514–528.
- [Eccles, 1958] Eccles, J. (1958). The physiology of nerve cells. *American Journal of Physical Medicine & Rehabilitation*, 37(6) :334.
- [Eichenbaum, 2017] Eichenbaum, H. (2017). Barlow versus hebb : When is it time to abandon the notion of feature detectors and adopt the cell assembly as the unit of cognition? *Neuroscience letters*.
- [Eker et al., 2004] Eker, S., Knapp, M., Laderoute, K., Lincoln, P., and Talcott, C. (2004). Pathway logic : Executable models of biological networks. *Electronic Notes in Theoretical Computer Science*, 71 :144–161.
- [Eliasmith et al., 2012] Eliasmith, C., Stewart, T., Choo, X., Bekolay, T., DeWolf, T., Tang, Y., and Rasmussen, D. (2012). A large-scale model of the functioning brain. *science*, 338(6111) :1202–1205.
- [Emerson, 1990] Emerson, E. (1990). Temporal and modal logic. In *Formal Models and Semantics*, pages 995–1072. Elsevier.
- [Engel et al., 1992] Engel, A., K3nig, P., Kreiter, A., Schillen, T., and Singer, W. (1992). Temporal coding in the visual cortex : new vistas on integration in the nervous system. *Trends Neurosci*, 15(6) :218–226.
- [EPFL, 2018] EPFL (2018). The blue brain project - a swiss brain initiative. <https://bluebrain.epfl.ch/page-52063.html>. [Page disponible en ligne en juin 2018].

- [Fages et al., 2004] Fages, F., Soliman, S., and Chabrier-Rivier, N. (2004). Modelling and querying interaction networks in the biochemical abstract machine biocham. *Journal of Biological Physics and Chemistry*, 4 :64–73.
- [Feldman, 2012] Feldman, D. (2012). The spike-timing dependence of plasticity. *Neuron*, 75(4) :556–571.
- [Filopon et al., 2006] Filopon, D., Mérieau, A., Bernot, G., Comet, J., LeBerre, R., Guery, B., Polack, B., and Guespin-Michel, J. (2006). Epigenetic acquisition of inducibility of type iii cytotoxicity in *p. aeruginosa*. *BMC bioinformatics*, 7(1) :272.
- [FitzHugh, 1961] FitzHugh, R. (1961). Impulses and physiological states in theoretical models of nerve membrane. *Biophysical journal*, 1(6) :445–466.
- [Fourcaud-Trocmé et al., 2003] Fourcaud-Trocmé, N., Hansel, D., Van Vreeswijk, C., and Brunel, N. (2003). How spike generation mechanisms determine the neuronal response to fluctuating inputs. *Journal of Neuroscience*, 23(37) :11628–11640.
- [Franzén, 2006] Franzén, A. (2006). Using satisfiability modulo theories for inductive verification of lustre programs. *Electronic Notes in Theoretical Computer Science*, 144(1) :19–33.
- [Froemke et al., 2010] Froemke, R., Letzkus, J., Kampa, B., Hang, G., and Stuart, G. (2010). Dendritic synapse location and neocortical spike-timing-dependent plasticity. *Frontiers in Synaptic Neuroscience*, 2 :29.
- [Fromentin et al., 2007] Fromentin, J., Comet, J., Le Gall, P., and Roux, O. (2007). Analysing gene regulatory networks by both constraint programming and model-checking. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 4595–4598. IEEE.
- [Fuortes et al., 1957] Fuortes, M., Frank, K., and Becker, M. (1957). Steps in the production of motoneuron spikes. *The Journal of general physiology*, 40(5) :735–752.
- [Gallet, 2016] Gallet, E. (2016). *Techniques de model-checking pour l'inférence de paramètres et l'analyse de réseaux biologiques*. PhD thesis, Université Paris-Saclay.
- [Galves and Löcherbach, 2013] Galves, A. and Löcherbach, E. (2013). Infinite systems of interacting chains with memory of variable length—a stochastic model for biological neural nets. *Journal of Statistical Physics*, 151(5) :896–921.
- [Gasparini et al., 2004] Gasparini, S., Migliore, M., and Magee, J. (2004). On the initiation and propagation of dendritic spikes in cal pyramidal neurons. *Journal of Neuroscience*, 24(49) :11046–11056.
- [Gautier et al., 1987] Gautier, T., Le Guernic, P., and Besnard, L. (1987). Signal : A declarative language for synchronous programming of real-time systems. In *Conference on Functional Programming Languages and Computer Architecture*, pages 257–277. Springer.
- [genesis, 2018] genesis (2018). <http://genesis-sim.org/>. [Page disponible en ligne en juillet 2018].

BIBLIOGRAPHIE

- [Gerstein and Kirkland, 2001] Gerstein, G. and Kirkland, K. (2001). Neural assemblies : technical issues, analysis, and modeling. *Neural Networks*, 14(6-7) :589–598.
- [Gerstner, 1995] Gerstner, W. (1995). Time structure of the activity in neural network models. *Physical review E*, 51(1) :738.
- [Gerstner et al., 1996] Gerstner, W., Kempter, R., van Hemmen, J., and Wagner, H. (1996). A neuronal learning rule for sub-millisecond temporal coding. *Nature*, 383(6595) :76.
- [Gerstner and Kistler, 2002] Gerstner, W. and Kistler, W. (2002). *Spiking neuron models : Single neurons, populations, plasticity*. Cambridge university press.
- [Gerstner et al., 2014] Gerstner, W., Kistler, W., Naud, R., and Paninski, L. (2014). *Neuronal dynamics : From single neurons to networks and models of cognition*. Cambridge University Press.
- [Gerstner and Naud, 2009] Gerstner, W. and Naud, R. (2009). How good are neuron models? *Science*, 326(5951) :379–380.
- [Gökçe et al., 2016] Gökçe, O., Bonhoeffer, T., and Scheuss, V. (2016). Clusters of synaptic inputs on dendrites of layer 5 pyramidal cells in mouse visual cortex. *Elife*, 5 :e09222.
- [goldman lab, 2018] goldman lab (2018). The integrate-and-fire model. http://neuroscience.ucdavis.edu/goldman/Tutorials_files/Integrate%26Fire.pdf. [Page disponible en ligne en juillet 2018].
- [Goncalvès da Silva, 2007] Goncalvès da Silva, E. (2007). Méthodes et analyse numériques.
- [Gorski et al., 2017] Gorski, T., Veltz, R., Galtier, M., Fragnaud, H., Telenczuk, B., and Destexhe, A. (2017). Inverse correlation processing by neurons with active dendrites. *bioRxiv*, page 137984.
- [Grammont and Riehle, 1999] Grammont, F. and Riehle, A. (1999). Precise spike synchronization in monkey motor cortex involved in preparation for movement. *Experimental brain research*, 128(1-2) :118–122.
- [Grammont and Riehle, 2003] Grammont, F. and Riehle, A. (2003). Spike synchronization and firing rate in a population of motor cortical neurons in relation to movement direction and reaction time. *Biological cybernetics*, 88(5) :360–373.
- [Gray, 1999] Gray, C. (1999). The temporal correlation hypothesis of visual feature integration : still alive and well. *Neuron*, 24(1) :31–47.
- [Gray et al., 1989] Gray, C., König, P., Engel, A., and Singer, W. (1989). Oscillatory responses in cat visual cortex exhibit inter-columnar synchronization which reflects global stimulus properties. *Nature*, 338(6213) :334.
- [Gray and Singer, 1989] Gray, C. and Singer, W. (1989). Stimulus-specific neuronal oscillations in orientation columns of cat visual cortex. *Proceedings of the National Academy of Sciences*, 86(5) :1698–1702.
- [Grothe, 2003] Grothe, B. (2003). New roles for synaptic inhibition in sound localization. *Nature Reviews Neuroscience*, 4(7) :540.

- [Grün et al., 1999] Grün, S., Diesmann, M., Grammont, F., Riehle, A., and Aertsen, A. (1999). Detecting unitary events without discretization of time. *Journal of neuroscience methods*, 94(1) :67–79.
- [Gunhanlar et al., 2017] Gunhanlar, N., Shpak, G., Van Der Kroeg, M., Gouty-Colomer, L., Munshi, S., Lendemeijer, B., Ghazvini, M., Dupont, C., Hoogendijk, W., Gribnau, J., et al. (2017). A simplified protocol for differentiation of electrophysiologically mature neuronal networks from human induced pluripotent stem cells. *Molecular psychiatry*.
- [Gurfinkel and Ivrii, 2017] Gurfinkel, A. and Ivrii, A. (2017). K-induction without unrolling. *Formal Methods in Computer-Aided Design FMCAD 2017*, page 148.
- [Guyonneau, 2006] Guyonneau, R. (2006). Codage par latence et stdp : des stratégies temporelles pour expliquer le traitement visuel rapide. *Unpublished thesis, Université Paul Sabatier, Toulouse, France*.
- [Guyonneau et al., 2005] Guyonneau, R., VanRullen, R., and Thorpe, S. (2005). Neurons tune to the earliest spikes through stdp. *Neural Computation*, 17(4) :859–879.
- [Hagen and Tinelli, 2008] Hagen, G. and Tinelli, C. (2008). Scaling up the formal verification of lustre programs with smt-based techniques. In *Formal Methods in Computer-Aided Design, 2008. FMCAD’08*, pages 1–9. IEEE.
- [Halbwachs, 1993] Halbwachs, N. (1993). *Synchronous Programming of Reactive Systems*. Kluwer Academic Publishers, Amsterdam.
- [Halbwachs, 2013] Halbwachs, N. (2013). *Synchronous programming of reactive systems*, volume 215. Springer Science & Business Media.
- [Halbwachs et al., 1991] Halbwachs, N., Caspi, P., Raymond, P., and Pilaud, D. (1991). The synchronous data flow programming language lustre. *Proceedings of the IEEE*, 79(9) :1305–1320.
- [Halbwachs et al., 1988] Halbwachs, N., Pilaud, D., Ouabdesselam, F., and Glory, A. (1988). Specifying, programming and verifying real-time systems using a synchronous declarative language. Technical Report L6, IMAG.
- [Harel and Pnueli, 1985] Harel, D. and Pnueli, A. (1985). On the development of reactive systems. In *Logics and models of concurrent systems*, pages 477–498. Springer.
- [Harris, 2005] Harris, K. (2005). Neural signatures of cell assembly organization. *Nature Reviews Neuroscience*, 6(5) :399.
- [Haykin, 2004] Haykin, S. (2004). A comprehensive foundation. *Neural networks*, 2(2004) :41.
- [Hebb, 1949] Hebb, D. (1949). The organization of behavior : A neurophysiological approach.
- [Hebb, 2002] Hebb, D. (2002). *The Organization of Behavior : A Neuropsychological Theory*. Taylor & Francis.
- [Heiner and Koch, 2004] Heiner, M. and Koch, I. (2004). Petri net based model validation in systems biology. In *International Conference on Application and Theory of Petri Nets*, pages 216–237. Springer.

BIBLIOGRAPHIE

- [Herculano-Houzel, 2009] Herculano-Houzel, S. (2009). The human brain in numbers : a linearly scaled-up primate brain. *Frontiers in human neuroscience*, 3 :31.
- [Hindmarsh and Rose, 1984] Hindmarsh, J. and Rose, R. (1984). A model of neuronal bursting using three coupled first order differential equations. *Proc. R. Soc. Lond. B*, 221(1222) :87–102.
- [Hines and Carnevale, 1997] Hines, M. and Carnevale, N. (1997). The neuron simulation environment. *Neural computation*, 9(6) :1179–1209.
- [Hodgkin and Huxley, 1952a] Hodgkin, A. and Huxley, A. (1952a). The components of membrane conductance in the giant axon of loligo. *The Journal of physiology*, 116(4) :473–496.
- [Hodgkin and Huxley, 1952b] Hodgkin, A. and Huxley, A. (1952b). Currents carried by sodium and potassium ions through the membrane of the giant axon of loligo. *The Journal of physiology*, 116(4) :449–472.
- [Hodgkin and Huxley, 1952c] Hodgkin, A. and Huxley, A. (1952c). The dual effect of membrane potential on sodium conductance in the giant axon of loligo. *The Journal of physiology*, 116(4) :497–506.
- [Hodgkin and Huxley, 1952d] Hodgkin, A. and Huxley, A. (1952d). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4) :500–544.
- [Hodgkin et al., 1952] Hodgkin, A., Huxley, A., and Katz, B. (1952). Measurement of current-voltage relations in the membrane of the giant axon of loligo. *The Journal of physiology*, 116(4) :424–448.
- [Hodgkin and Rushton, 1946] Hodgkin, A. and Rushton, W. (1946). The electrical constants of a crustacean nerve fibre. *Proc. R. Soc. Lond. B*, 133(873) :444–479.
- [Holmes, 2013] Holmes, W. (2013). *Cable Equation*, pages 1–13. Springer New York, New York, NY.
- [Holmes et al., 1992] Holmes, W., Segev, I., and Rall, W. (1992). Interpretation of time constant and electrotonic length estimates in multicylinder or branched neuronal structures. *Journal of Neurophysiology*, 68(4) :1401–1420.
- [Holtmaat and Caroni, 2016] Holtmaat, A. and Caroni, P. (2016). Functional and structural underpinnings of neuronal assembly formation in learning. *Nature neuroscience*, 19(12) :1553.
- [Horst, 2011] Horst, S. (2011). The computational theory of mind. *Stanford Encyclopedia of Philosophy*.
- [Hubel and Wiesel, 1962] Hubel, D. and Wiesel, T. (1962). Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex. *The Journal of physiology*, 160(1) :106–154.
- [Huth and Ryan, 2004] Huth, M. and Ryan, M. (2004). *Logic in Computer Science : Modelling and reasoning about systems*. Cambridge university press.

-
- [IBM, 2012] IBM (2012). Blue gene. <http://www-03.ibm.com/ibm/history/ibm100/us/en/icons/bluegene/>. [Page disponible en ligne en juin 2018].
- [inria saclay-ile de france, 2018] inria saclay-ile de france, E. L. (2018). The biochemical abstract machine biocham 3. <http://lifeware.inria.fr/biocham/>. [Page disponible en ligne en juillet 2018].
- [IRCCyN, 2018] IRCCyN (2018). Roméo - a tool for time petri nets analysis. http://romeo.rts-software.org/?page_id=2. [Page disponible en ligne en juillet 2018].
- [Izhikevich, 2001] Izhikevich, E. (2001). Resonate-and-fire neurons. *Neural networks*, 14(6-7) :883–894.
- [Izhikevich, 2003] Izhikevich, E. (2003). Simple model of spiking neurons. *IEEE Transactions on neural networks*, 14(6) :1569–1572.
- [Izhikevich, 2004] Izhikevich, E. (2004). Which model to use for cortical spiking neurons? *IEEE transactions on neural networks*, 15(5) :1063–1070.
- [Izhikevich, 2006] Izhikevich, E. (2006). Polychronization : computation with spikes. *Neural computation*, 18(2) :245–282.
- [Jeannet, 2000] Jeannet, B. (2000). *Partitionnement Dynamique dans l'Analyse de Relations Linéaires et Application à la Vérification de Programmes Synchrones*. PhD thesis, Ecole Doctorale Mathématiques, Sciences et technologie de l'Information.
- [Jeannet, 2010] Jeannet, B. (2010). The NBAC verification. <http://pop-art.inrialpes.fr/people/bjeannet/nbac/index.html>. [Page disponible en ligne en juin 2018].
- [Kahsai and Tinelli, 2011] Kahsai, T. and Tinelli, C. (2011). Pkind : A parallel k-induction based model checker. *arXiv preprint arXiv :1111.0372*.
- [Kaiser and Angell, 1957] Kaiser, J. and Angell, R. (1957). *New techniques and equipment for correlation and computation*. Servomechanisms Laboratory, Dept. of Electrical Engineering, Massachusetts Institute of Technology.
- [Kandel et al., 2012] Kandel, E., Schwartz, J., Jessell, T., Siegelbaum, S., Hudspeth, A., et al. (2012). *Principles of neural science*, volume 5. McGraw-hill New York.
- [Katona et al., 2011] Katona, G., Kaszás, A., Turi, G., Hájos, N., Tamás, G., Vizi, E., and Rózsa, B. (2011). Roller coaster scanning reveals spontaneous triggering of dendritic spikes in cal interneurons. *Proceedings of the National Academy of Sciences*, 108(5) :2148–2153.
- [Klug and Albrecht, 2015] Klug, A. and Albrecht, O. (2015). Neural circuits : Introducing different scales of temporal processing. *Current Biology*, 25(13).
- [Koch, 2004] Koch, C. (2004). *Biophysics of computation : information processing in single neurons*. Oxford university press.
- [Koch et al., 1990] Koch, C., Douglas, R., and Wehmeier, U. (1990). Visibility of synaptically induced conductance changes : theory and simulations of anatomically characterized cortical pyramidal cells. *Journal of Neuroscience*, 10(6) :1728–1744.

BIBLIOGRAPHIE

- [Koch et al., 1982] Koch, C., Poggio, T., and Torre, V. (1982). Retinal ganglion cells : a functional interpretation of dendritic morphology. *Phil. Trans. R. Soc. Lond. B*, 298(1090) :227–263.
- [Koch and Segev, 1998] Koch, C. and Segev, I. (1998). *Methods in neuronal modeling : from ions to networks*. MIT press.
- [Koch et al., 2004] Koch, I., Junker, B., and Heiner, M. (2004). Application of petri net theory for modelling and validation of the sucrose breakdown pathway in the potato tuber. *Bioinformatics*, 21(7) :1219–1226.
- [Kohn, 1999] Kohn, K. (1999). Molecular interaction map of the mammalian cell cycle control and dna repair systems. *Molecular biology of the cell*, 10(8) :2703–2734.
- [König et al., 1996] König, P., Engel, A., and Singer, W. (1996). Integrator or coincidence detector ? the role of the cortical neuron revisited. *Trends in neurosciences*, 19(4) :130–137.
- [Köppl, 2012] Köppl, C. (2012). Auditory neuroscience : how to encode microsecond differences. *Current Biology*, 22(2) :R56–R58.
- [Kullmann et al., 2012] Kullmann, D., Moreau, A., Bakiri, Y., and Nicholson, E. (2012). Plasticity of inhibition. *Neuron*, 75(6) :951–962.
- [Kupferschmidt, 2015] Kupferschmidt, K. (2015). Virtual rat brain fails to impress its critics.
- [Kurzweil, 2011] Kurzweil (2011). Ibm announces 20-petaflops supercomputer. <http://www.kurzweilai.net/ibm-announces-20-petaflops-supercomputer>. [Page disponible en ligne en juin 2018].
- [LAAS, 2018] LAAS (2018). Time petri net analyzer. <http://projects.laas.fr/tina/home.php>. [Page disponible en ligne en juillet 2018].
- [Lamport, 1977] Lamport, L. (1977). Proving the correctness of multiprocess programs. *IEEE transactions on software engineering*, (2) :125–143.
- [Lapicque, 1907] Lapicque, L. (1907). Recherches quantitatives sur l’excitation électrique des nerfs traitée comme une polarisation. *Journal de physiologie et de pathologie générale*, 9 :620–635.
- [le journal international, 2013] le journal international (2013). Suisse : le human brain project, ou le fantasme de l’immortalité digitale. <http://www.lejournalinternational.info/suisse-human-brain-project-fantasme-de-limmortalite-digitale/>. [Page disponible en ligne en juin 2018].
- [Leshno et al., 1993] Leshno, M., Lin, V., Pinkus, A., and Schocken, S. (1993). Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks*, 6(6) :861–867.
- [Lewicki, 1998] Lewicki, M. (1998). A review of methods for spike sorting : the detection and classification of neural action potentials. *Network : Computation in Neural Systems*, 9(4) :R53–R78.

- [Losonczy and Magee, 2006] Losonczy, A. and Magee, J. (2006). Integrative properties of radial oblique dendrites in hippocampal cal pyramidal neurons. *Neuron*, 50(2) :291–307.
- [Lowel and Singer, 1992] Lowel, S. and Singer, W. (1992). Selection of intrinsic horizontal connections in the visual cortex by correlated neuronal activity. *Science*, 255(5041) :209–212.
- [ls2n, 2018] ls2n (2018). laboratoire des sciences du numérique de nantes. <https://www.ls2n.fr/>. [Page disponible en ligne en juillet 2018].
- [Maass, 1997] Maass, W. (1997). Networks of spiking neurons : the third generation of neural network models. *Neural networks*, 10(9) :1659–1671.
- [Maass, 1999] Maass, W. (1999). Computing with spiking neurons. *Pulsed neural networks*, 85.
- [Maass and Bishop, 2001] Maass, W. and Bishop, C. (2001). *Pulsed neural networks*. MIT press.
- [Magee and Cook, 2000] Magee, J. and Cook, E. (2000). Somatic epsp amplitude is independent of synapse location in hippocampal pyramidal neurons. *Nature neuroscience*, 3(9) :895.
- [Major et al., 1993] Major, G., Evans, J., and Jack, J. (1993). Solutions for transients in arbitrarily branching cables : I. voltage recording with a somatic shunt. *Biophysical Journal*, 65(1) :423–449.
- [Malvache et al., 2016] Malvache, A., Reichinnek, S., Villette, V., Haimerl, C., and Cossart, R. (2016). Awake hippocampal reactivations project onto orthogonal neuronal assemblies. *Science*, 353(6305) :1280–1283.
- [Mann and Paulsen, 2007] Mann, E. and Paulsen, O. (2007). Role of gabaergic inhibition in hippocampal network oscillations. *Trends in neurosciences*, 30(7) :343–349.
- [Markham and Zakon, 2014] Markham, M. and Zakon, H. (2014). Ionic mechanisms of microsecond-scale spike timing in single cells. *Journal of Neuroscience*, 34(19) :6668–6678.
- [Markram, 2006] Markram, H. (2006). The blue brain project. *Nature Reviews Neuroscience*, 7(2) :153.
- [Markram et al., 1997] Markram, H., Lübke, J., Frotscher, M., and Sakmann, B. (1997). Regulation of synaptic efficacy by coincidence of postsynaptic aps and epsps. *Science*, 275(5297) :213–215.
- [Markram et al., 2015] Markram, H., Muller, E., Ramaswamy, S., Reimann, M., Abdellah, M., Sanchez, C., Ailamaki, A., Alonso-Nanclares, L., Antille, N., Arsever, S., et al. (2015). Reconstruction and simulation of neocortical microcircuitry. *Cell*, 163(2) :456–492.
- [Martinez and Paugam-Moisy, 2008] Martinez, R. and Paugam-Moisy, H. (2008). Les groupes polychrones pour capturer l’aspect spatio-temporel de la mémorisation. In *Deuxième conférence française de Neurosciences Computationnelles, " Neurocomp08"*.

BIBLIOGRAPHIE

- [Masland, 2004] Masland, R. (2004). Neuronal cell types. *Current Biology*, 14(13) :R497–R500.
- [Masquelier et al., 2008] Masquelier, T., Guyonneau, R., and Thorpe, S. (2008). Spike timing dependent plasticity finds the start of repeating patterns in continuous spike trains. *PloS one*, 3(1) :e1377.
- [Masquelier and Thorpe, 2007] Masquelier, T. and Thorpe, S. (2007). Unsupervised learning of visual features through spike timing dependent plasticity. *PLoS computational biology*, 3(2) :e31.
- [Matignon, 2005] Matignon, R. (2005). *Neural Network Modeling Using Sas Enterprise Miner*. AuthorHouse.
- [Matsuzaki et al., 2001] Matsuzaki, M., Ellis-Davies, G., Nemoto, T., Miyashita, Y., Iino, M., and Kasai, H. (2001). Dendritic spine geometry is critical for ampa receptor expression in hippocampal ca1 pyramidal neurons. *Nature neuroscience*, 4(11) :1086.
- [McCulloch and Pitts, 1943] McCulloch, W. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4) :115–133.
- [McNaughton et al., 1981] McNaughton, B., Barnes, C., and Andersen, P. (1981). Synaptic efficacy and epsp summation in granule cells of rat fascia dentata studied in vitro. *Journal of Neurophysiology*, 46(5) :952–966.
- [Mebout, 2014] Mebout, A. (2014). *Inférence d’invariants pour le model checking de systèmes paramétrés*. PhD thesis, Université Paris Sud-Paris XI.
- [Megias et al., 2001] Megias, M., Emri, Z., Freund, T., and Gulyas, A. (2001). Total number and distribution of inhibitory and excitatory synapses on hippocampal ca1 pyramidal cells. *Neuroscience*, 102(3) :527–540.
- [Meister et al., 1995] Meister, M., Lagnado, L., and Baylor, D. (1995). Concerted signaling by retinal ganglion cells. *Science*, 270(5239) :1207–1210.
- [Mel, 1992a] Mel, B. (1992a). The clusteron : toward a simple abstraction for a complex neuron. In *Advances in neural information processing systems*, pages 35–42.
- [Mel, 1992b] Mel, B. (1992b). Nmda-based pattern discrimination in a modeled cortical neuron. *Neural Computation*, 4(4) :502–517.
- [Mel, 1994] Mel, B. (1994). Information processing in dendritic trees. *Neural computation*, 6(6) :1031–1085.
- [Meunier et al., 2006] Meunier, D., Martinez, R., and Paugam-Moisy, H. (2006). Assemblées temporelles dans les réseaux de neurones impulsifs. *Proc. of NeuroComp*, 6 :187–190.
- [Miles and Wong, 1986] Miles, R. and Wong, R. (1986). Excitatory synaptic interactions between ca3 neurones in the guinea-pig hippocampus. *The Journal of Physiology*, 373(1) :397–418.
- [Minsky and Papert, 1969] Minsky, M. and Papert, S. (1969). *Perceptrons : An Introduction to computational geometry*.

- [Minta et al., 1989] Minta, A., Kao, J., and Tsien, R. (1989). Fluorescent indicators for cytosolic calcium based on rhodamine and fluorescein chromophores. *Journal of Biological Chemistry*, 264(14) :8171–8178.
- [Miraglio et al., 2016] Miraglio, B., Bernot, G., Comet, J., and Risso de Faverney, C. (2016). Towards a computer aided toxicology. *advances in Systems and Synthetic Biology*.
- [Misawa et al., 2004] Misawa, S., Kuwabara, S., Ogawara, K., Kitano, Y., Yagui, K., and Hattori, T. (2004). Hyperglycemia alters refractory periods in human diabetic neuropathy. *Clinical neurophysiology*, 115(11) :2525–2529.
- [Mishra et al., 2006] Mishra, D., Yadav, A., and Kalra, P. (2006). Learning with single quadratic integrate-and-fire neuron. In *International Symposium on Neural Networks*, pages 424–429. Springer.
- [Mitchison, 1991] Mitchison, G. (1991). Neuronal branching patterns and the economy of cortical wiring. *Proc. R. Soc. Lond. B*, 245(1313) :151–158.
- [Morris and Lecar, 1981] Morris, C. and Lecar, H. (1981). Voltage oscillations in the barnacle giant muscle fiber. *Biophysical journal*, 35(1) :193–213.
- [Mostafa, 2018] Mostafa, H. (2018). Supervised learning based on temporal coding in spiking neural networks. *IEEE transactions on neural networks and learning systems*, 29(7) :3227–3235.
- [Murata, 1989] Murata, T. (1989). Petri nets : properties, analysis and applications. *Proceedings of the IEEE*, 77(4) :541–580.
- [Muzy et al., 2017] Muzy, A., Zeigler, B., and Grammont, F. (2017). Iterative specification as a modeling and simulation formalism for i/o general systems. *IEEE Systems Journal*.
- [Nair and Hinton, 2010] Nair, V. and Hinton, G. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.
- [Nedjah et al., 2011] Nedjah, N., Coelho, L., Mariani, V., and de Macedo Mourelle, L. (2011). *Innovative Computing Methods and Their Applications to Engineering Problems*. Studies in Computational Intelligence. Springer.
- [Nelson et al., 2006] Nelson, S., Sugino, K., and Hempel, C. (2006). The problem of neuronal cell types : a physiological genomics approach. *Trends in neurosciences*, 29(6) :339–345.
- [neuron, 2018] neuron (2018). <https://neuron.yale.edu/neuron/>. [Page disponible en ligne en juillet 2018].
- [Nevian et al., 2007] Nevian, T., Larkum, M., Polsky, A., and Schiller, J. (2007). Properties of basal dendrites of layer 5 pyramidal neurons : a direct patch-clamp recording study. *Nature neuroscience*, 10(2) :206.
- [Nicolas, 2004] Nicolas, H. (2004). Résolution des Équations Différentielles. <http://maverick.inria.fr/~Nicolas.Holzschuch/cours/class5.pdf>. [Page disponible en ligne en juin 2018].

BIBLIOGRAPHIE

- [Nicoletis et al., 1997] Nicoletis, M., Fanselow, E., and Ghazanfar, A. (1997). Hebb’s dream : the resurgence of cell assemblies. *Neuron*, 19(2) :219–221.
- [Nielsen, 2015] Nielsen, M. (2015). *Neural networks and deep learning*. Determination Press.
- [Nordlie et al., 2010] Nordlie, E., Tetzlaff, T., and Einevoll, G. (2010). Rate dynamics of leaky integrate-and-fire neurons with strong synapses. *Frontiers in computational neuroscience*, 4 :149.
- [Nossenson and Messer, 2010] Nossenson, N. and Messer, H. (2010). Modeling neuron firing pattern using a two state markov chain. In *Sensor Array and Multichannel Signal Processing Workshop (SAM), 2010 IEEE*, pages 41–44. IEEE.
- [Notari, 2009] Notari, C. (2009). *Métaphore de l’ordinateur et linguistique cognitive*. PhD thesis, ANRT.
- [Orhan, 2012] Orhan, E. (2012). The leaky integrate-and-fire neuron model. *no*, 3 :1–6.
- [Pace et al., 2004] Pace, G., Halbwachs, N., and Raymond, P. (2004). Counter-example generation in symbolic abstract model-checking. *International Journal on Software Tools for Technology Transfer*, 5(2-3) :158–164.
- [Petreanu et al., 2009] Petreanu, L., Mao, T., Sternson, S., and Svoboda, K. (2009). The subcellular organization of neocortical excitatory connections. *Nature*, 457(7233) :1142.
- [Petri, 1980] Petri, C. (1980). Introduction to general net theory. In Brauer, W., editor, *Net theory and Applications*, volume 84 of *Lecture Notes in Computer Science*, pages 1–19. Springer-Verlag.
- [Peytavin, 1987] Peytavin, J. (1987). Quelques réflexions sur les modèles du cerveau. *Qua-derni*, 1(1) :27–33.
- [Pierce and Benjamin, 2002] Pierce, B. and Benjamin, C. (2002). *Types and programming languages*. MIT press.
- [Pilaud et al., 1987] Pilaud, D., Halbwachs, N., and Plaice, J. (1987). Lustre : A declarative language for programming synchronous systems. In *Proceedings of the 14th Annual ACM Symposium on Principles of Programming Languages (14th POPL 1987)*. ACM, New York, NY, volume 178, page 188.
- [Platkiewicz and Brette, 2010] Platkiewicz, J. and Brette, R. (2010). A threshold equation for action potential initiation. *PLoS computational biology*, 6(7) :e1000850.
- [Poirazi et al., 2003] Poirazi, P., Brannon, T., and Mel, B. (2003). Pyramidal neuron as two-layer neural network. *Neuron*, 37(6) :989–999.
- [Poirazi and Mel, 2001] Poirazi, P. and Mel, B. (2001). Impact of active dendrites and structural plasticity on the memory capacity of neural tissue. *Neuron*, 29(3) :779–796.
- [Popper, 2014] Popper, K. (2014). *Conjectures and refutations : The growth of scientific knowledge*. routledge.
- [Press et al., 1996] Press, W., Teukolsky, S., Vetterling, W., and Flannery, B. (1996). *Numerical recipes in C*, volume 2. Cambridge university press Cambridge.

- [Priddy and Keller, 2005] Priddy, K. and Keller, P. (2005). *Artificial Neural Networks : An Introduction*. SPIE tutorial texts. Society of Photo Optical.
- [Pritchard et al., 2002] Pritchard, T., Alloway, K., and Dinh, Y. (2002). *Neurosciences médicales : Les bases neuroanatomiques et neurophysiologiques*. Sciences médicales. De Boeck Supérieur.
- [Puppo et al., 2018] Puppo, F., George, V., and Silva, G. (2018). An optimized structure-function design principle underlies efficient signaling dynamics in neurons. *Scientific reports*, 8(1) :10460.
- [Purves et al., 2001a] Purves, D., Augustine, G., and Fitzpatrick, D. (2001a). *Neuroscience. 2nd edition*. Sunderland (MA) : Sinauer Associates.
- [Purves et al., 2001b] Purves, D., Augustine, G., Fitzpatrick, D., Katz, L., LaMantia, A., McNamara, J., and Williams, S. (2001b). Increased conduction velocity as a result of myelination. *Neuroscience*.
- [Rall, 1957] Rall, W. (1957). Membrane time constant of motoneurons. *Science (New York, NY)*, 126(3271) :454.
- [Rall, 1959] Rall, W. (1959). Branching dendritic trees and motoneuron membrane resistivity. *Experimental neurology*, 1(5) :491–527.
- [Rall, 1962] Rall, W. (1962). Theory of physiological properties of dendrites. *Annals of the New York Academy of Sciences*, 96(1) :1071–1092.
- [Rall, 1969] Rall, W. (1969). Time constants and electrotonic length of membrane cylinders and neurons. *Biophysical Journal*, 9(12) :1483–1508.
- [Rall, 2011] Rall, W. (2011). Core conductor theory and cable properties of neurons. *Comprehensive physiology*.
- [Rall and Rinzel, 1973] Rall, W. and Rinzel, J. (1973). Branch input resistance and steady attenuation for input to one branch of a dendritic neuron model. *Biophysical journal*, 13(7) :648–688.
- [Ratnadurai Giridharan et al., 2014] Ratnadurai Giridharan, S., Stefanescu, R., Khargonekar, P., Carney, P., and Talathi, S. (2014). Genesis of interictal spikes in the ca1 : a computational investigation. *Frontiers in neural circuits*, 8 :2.
- [Raymond, 2008] Raymond, P. (2008). Synchronous program verification with lustre/lesar. *Modeling and Verification of Real-Time Systems*, page 7.
- [Reddy et al., 1993] Reddy, V., Mavrovouniotis, M., Liebman, M., et al. (1993). Petri net representations in metabolic pathways. In *ISMB*, volume 93, pages 328–336.
- [Regev et al., 2004] Regev, A., Panina, E., Silverman, W., Cardelli, L., and Shapiro, E. (2004). Bioambients : an abstraction for biological compartments. *Theoretical Computer Science*, 325(1) :141–167.
- [Reynaud-Bouret et al., 2014] Reynaud-Bouret, P., Rivoirard, V., Grammont, F., and Tuleau-Malot, C. (2014). Goodness-of-fit tests and nonparametric adaptive estimation for spike train analysis. *The Journal of Mathematical Neuroscience*, 4(1) :3.

BIBLIOGRAPHIE

- [Richard, 2010] Richard, A. (2010). Smbionet user manual.
- [Riehle et al., 2000] Riehle, A., Grammont, F., Diesmann, M., and Grün, S. (2000). Dynamical changes and temporal precision of synchronized spiking activity in monkey motor cortex during movement preparation. *Journal of Physiology-Paris*, 94(5-6) :569–582.
- [Rinzel and Rall, 1974] Rinzel, J. and Rall, W. (1974). Transient response in a dendritic neuron model for current injected at one branch. *Biophysical Journal*, 14(10) :759–790.
- [Rosenblatt, 1958] Rosenblatt, F. (1958). The perceptron : a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6) :386.
- [Rossant et al., 2011] Rossant, C., Goodman, D., Fontaine, B., Platkiewicz, J., Magnusson, A., and Brette, R. (2011). Fitting neuron models to spike trains. *Frontiers in neuroscience*, 5 :9.
- [Rumelhart et al., 1985] Rumelhart, D., Hinton, G., and Williams, R. (1985). Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.
- [Sakurai, 1999] Sakurai, Y. (1999). How do cell assemblies encode information in the brain ? *Neuroscience & Biobehavioral Reviews*, 23(6) :785–796.
- [Salinas and Sejnowski, 2002] Salinas, E. and Sejnowski, T. (2002). Integrate-and-fire neurons driven by correlated stochastic input. *Neural computation*, 14(9) :2111–2155.
- [Schwiening, 2012] Schwiening, C. (2012). A brief historical perspective : Hodgkin and huxley. *The Journal of physiology*, 590(11) :2571–2575.
- [Sebastian, 2014] Sebastian, W. (2014). Model Checking with IC3. https://www.tcs.cs.tu-bs.de/documents/Logics_Seminar_2014/IC3.pdf. [Page disponible en ligne en juin 2018].
- [Segev et al., 1994] Segev, I., Rinzel, J., and Shepherd, G. (1994). *The theoretical foundation of dendritic function*. Cambridge, MA : The MIT Press.
- [Shazeer et al., 2017] Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., and Dean, J. (2017). Outrageously large neural networks : The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv :1701.06538*.
- [Shepherd and Brayton, 1987] Shepherd, G. and Brayton, R. (1987). Logic operations are properties of computer-simulated interactions between excitable dendritic spines. *Neuroscience*, 21(1) :151–165.
- [Shepherd, 2004] Shepherd, G. M. (2004). In Byrne, J. and Roberts, J., editors, *From Molecules to Networks*, pages 91 – 113. Academic Press, Burlington.
- [Sheppard and Wilson, 1981] Sheppard, C. and Wilson, T. (1981). The theory of the direct-view confocal microscope. *Journal of microscopy*, 124(2) :107–117.
- [Siebert, 1965] Siebert, W. (1965). Some implications of the stochastic behavior of primary auditory neurons. *Kybernetik*, 2(5) :206–215.
- [Singer, 1999] Singer, W. (1999). Neuronal synchrony : a versatile code for the definition of relations ? *Neuron*, 24(1) :49–65.

- [Singer, 2007] Singer, W. (2007). Understanding the brain : How can our intuition fail so fundamentally when it comes to studying the organ to which it owes its existence ? *EMBO reports*, 8(1S) :S16–S19.
- [Singer et al., 1997] Singer, W., Engel, A., Kreiter, A., Munk, M., Neuenschwander, S., and Roelfsema, P. (1997). Neuronal assemblies : necessity, signature and detectability. *Trends in cognitive sciences*, 1(7) :252–261.
- [Singer and Gray, 1995] Singer, W. and Gray, C. (1995). Visual feature integration and the temporal correlation hypothesis. *Annual review of neuroscience*, 18(1) :555–586.
- [Sjöström and Gerstner, 2010] Sjöström, J. and Gerstner, W. (2010). Spike-timing dependent plasticity. *Scholarpedia*, 5(2) :1362. revision #184913.
- [Snoussi, 1989] Snoussi, E. (1989). Qualitative dynamics of piecewise-linear differential equations : a discrete mapping approach. *Dynamics and stability of Systems*, 4(3-4) :565–583.
- [Spruston, 2008] Spruston, N. (2008). Pyramidal neurons : dendritic structure and synaptic integration. *Nature Reviews Neuroscience*, 9(3) :206.
- [Starke, 2000] Starke, P. (2000). Ina integrated net analyzer. <https://www2.informatik.hu-berlin.de/~starke/ina.html>. [Page disponible en ligne en juillet 2018].
- [Stevens and Zador, 1995] Stevens, C. and Zador, A. (1995). Neural coding : The enigma of the brain. *Current Biology*, 5(12) :1370–1371.
- [Stuart and Sakmann, 1994] Stuart, G. and Sakmann, B. (1994). Active propagation of somatic action potentials into neocortical pyramidal cell dendrites. *Nature*, 367(6458) :69.
- [Stuart and Spruston, 1998] Stuart, G. and Spruston, N. (1998). Determinants of voltage attenuation in neocortical pyramidal neuron dendrites. *Journal of Neuroscience*, 18(10) :3501–3510.
- [Stuart and Spruston, 2015] Stuart, G. and Spruston, N. (2015). Dendritic integration : 60 years of progress. *Nature neuroscience*, 18(12) :1713.
- [Stuart et al., 2016] Stuart, G., Spruston, N., and Häusser, M. (2016). *Dendrites*. Oxford University Press.
- [Tavanaei et al., 2018] Tavanaei, A., Ghodrati, M., Kheradpisheh, S., Masquelier, T., and Maida, A. (2018). Deep learning in spiking neural networks. *arXiv preprint arXiv :1804.08150*.
- [technologies clés 2020, 2018] technologies clés 2020 (2018). Supercalculateurs. <https://technologies-cles-2020.entreprises.gouv.fr/assets/pdf/technologies/technologie-21.pdf>. [Page disponible en ligne en juin 2018].
- [Theunissen and Miller, 1995] Theunissen, F. and Miller, J. (1995). Temporal encoding in nervous systems : a rigorous definition. *Journal of computational neuroscience*, 2(2) :149–162.
- [Thiré, 2016] Thiré, F. (2016). Partie 1 : SAT solveur. <http://www.lsv.fr/~fthire/teaching/2016-2017/projet-logique/OCaml/sujet/sujet.pdf>. [Page disponible en ligne en juin 2018].

BIBLIOGRAPHIE

- [Thomas, 1978] Thomas, R. (1978). Logical analysis of systems comprising feedback loops. *Journal of Theoretical Biology*, 73(4) :631–656.
- [Thomas, 1991] Thomas, R. (1991). Regulatory networks seen as asynchronous automata : a logical description. *Journal of theoretical biology*, 153(1) :1–23.
- [Thomas and d’Ari, 1990] Thomas, R. and d’Ari, R. (1990). *Biological feedback*. CRC press.
- [Thomas and Kaufman, 2001] Thomas, R. and Kaufman, M. (2001). Multistationarity, the basis of cell differentiation and memory. i. structural conditions of multistationarity and other nontrivial behavior. *Chaos : An Interdisciplinary Journal of Nonlinear Science*, 11(1) :170–179.
- [Thomas et al., 1995] Thomas, R., Thieffry, D., and Kaufman, M. (1995). Dynamical behaviour of biological regulatory networks—i. biological role of feedback loops and practical use of the concept of the loop-characteristic state. *Bulletin of mathematical biology*, 57(2) :247–276.
- [Thomson, 1856] Thomson, W. (1856). On the theory of the electric telegraph. *Proceedings of the Royal Society of London*, 7 :382–399.
- [Thorpe, 1990] Thorpe, S. (1990). Spike arrival times : A highly efficient coding scheme for neural networks. *Parallel processing in neural systems*, pages 91–94.
- [Thorpe et al., 2001] Thorpe, S., Delorme, A., and Van Rullen, R. (2001). Spike-based strategies for rapid processing. *Neural networks*, 14(6-7) :715–725.
- [Thorpe and Imbert, 1989] Thorpe, S. and Imbert, M. (1989). Biological constraints on connectionist modelling. *Connectionism in perspective*, pages 63–92.
- [Thorson, 1966] Thorson, J. (1966). Small-signal analysis of a visual reflex in the locust. *Kybernetik*, 3(2) :53–66.
- [Tiwari et al., 2001] Tiwari, A., Rueß, H., Saïdi, H., and Shankar, N. (2001). A technique for invariant generation. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 113–127. Springer.
- [TOP500, 2018] TOP500 (2018). Home - top500 supercomputer sites. <https://www.top500.org>. [Page disponible en ligne en juin 2018].
- [Touboul, 2009] Touboul, J. (2009). Importance of the cutoff value in the quadratic adaptive integrate-and-fire model. *Neural computation*, 21(8) :2114–2122.
- [Touzet, 1992] Touzet, C. (1992). *Les réseaux de neurones artificiels, introduction au connexionnisme*. EC2.
- [Troncale et al., 2006] Troncale, S., Tahi, F., Campard, D., Vannier, J., and Guespin, J. (2006). Modeling and simulation with hybrid functional petri nets of the role of interleukin-6 in human early haematopoiesis. In *Biocomputing 2006*, pages 427–438. World Scientific.
- [Truccolo et al., 2011] Truccolo, W., Donoghue, J., Hochberg, L., Eskandar, E., Madsen, J., Anderson, W., Brown, E., Halgren, E., and Cash, S. (2011). Single-neuron dynamics in human focal epilepsy. *Nature neuroscience*, 14(5) :635.

- [Tuckwell, 1988] Tuckwell, H. (1988). Introduction to theoretical neurobiology. vol. 1, linear cable theory and dendritic structure and stochastic theories.
- [Tuleau-Malot et al., 2014] Tuleau-Malot, C., Rouis, A., Grammont, F., and Reynaud-Bouret, P. (2014). Multiple tests based on a gaussian approximation of the unitary events method with delayed coincidence count. *Neural computation*, 26(7) :1408–1454.
- [Tuncer et al., 2017] Tuncer, S., Peker, T., Burat, I., Kiziltan, E., İlhan, B., and Dalkılıç, N. (2017). Axonal excitability and conduction alterations caused by levobupivacaine in rat. *Acta Pharmaceutica*, 67(3) :293–307.
- [uppaal, 2015] uppaal (2015). <http://www.uppaal.org/>. [Page disponible en ligne en juillet 2018].
- [Usher et al., 1993] Usher, M., Schuster, H., and Niebur, E. (1993). Dynamics of populations of integrate-and-fire neurons, partial synchronization and memory. *Neural Computation*, 5(4) :570–586.
- [Usrey and Reid, 1999] Usrey, W. and Reid, R. (1999). Synchronous activity in the visual system. *Annual review of physiology*, 61(1) :435–456.
- [Van Rossum, 2014] Van Rossum, M. (2014). Neural computation msc course - practical 4 : Integrate and fire neurons. <http://www.inf.ed.ac.uk/teaching/courses/nc/IF.pdf>. [Page disponible en ligne en juillet 2018].
- [Verimag, 2018] Verimag (2018). Formal Verification, Theory, Techniques and Tools. <http://www-verimag.imag.fr/Formal-Verification-Theory.html?lang=fr>. [Page disponible en ligne en juin 2018].
- [Von der Malsburg, 1987] Von der Malsburg, C. (1987). Synaptic plasticity as basis of brain organization. *The neural and molecular bases of learning*, 411 :432.
- [Von der Malsburg, 1995] Von der Malsburg, C. (1995). Binding in models of perception and brain function. *Current opinion in neurobiology*, 5(4) :520–526.
- [Von der Malsburg, 1999] Von der Malsburg, C. (1999). The what and why of binding : the modeler’s perspective. *Neuron*, 24(1) :95–104.
- [Wendling et al., 2002] Wendling, F., Bartolomei, F., Bellanger, J., and Chauvel, P. (2002). Epileptic fast activity can be explained by a model of impaired gabaergic dendritic inhibition. *European Journal of Neuroscience*, 15(9) :1499–1508.
- [Widrow and Lehr, 1990] Widrow, B. and Lehr, M. (1990). 30 years of adaptive neural networks : perceptron, madaline, and backpropagation. *Proceedings of the IEEE*, 78(9) :1415–1442.
- [Wikipédia, 2018] Wikipédia (2018). Fonction d’activation — wikipédia, l’encyclopédie libre. http://fr.wikipedia.org/w/index.php?title=Fonction_d%27activation&oldid=149507481. [Page disponible en ligne en juin 2018].
- [Williams and Stuart, 2002] Williams, S. and Stuart, G. (2002). Dependence of epsp efficacy on synapse location in neocortical pyramidal neurons. *Science*, 295(5561) :1907–1910.

BIBLIOGRAPHIE

- [WoPeD, 2018] WoPeD (2018). Workflow petri net designer. <https://woped.dhbw-karlsruhe.de/>. [Page disponible en ligne en juillet 2018].
- [Wu and Mel, 2009] Wu, X. and Mel, B. (2009). Capacity-enhancing synaptic learning rules in a medial temporal lobe online learning model. *Neuron*, 62(1) :31–41.
- [Yoshua, 2018] Yoshua, B. (2018). Introduction aux algorithmes d’apprentissage profonds. http://www.iro.umontreal.ca/~bengioy/ift6266/H12/html/deepintro_fr.html#cognition-fr. [Page disponible en ligne en juin 2018].
- [Young, 1936] Young, J. (1936). Structure of nerve fibres and synapses in some invertebrates. In *Cold Spring Harbor Symposia on Quantitative Biology*, volume 4, pages 1–6. Cold Spring Harbor Laboratory Press.
- [Yuste and Tank, 1996] Yuste, R. and Tank, D. (1996). Dendritic integration in mammalian neurons, a century after cajal. *Neuron*, 16(4) :701–716.
- [Zadeh and Kambhampati, 2017] Zadeh, S. and Kambhampati, C. (2017). All-or-none principle and weakness of hodgkin-huxley mathematical model. *World Academy of Science, Engineering and Technology, International Journal of Mathematical, Computational, Physical, Electrical and Computer Engineering*, 11(11) :467–471.