

# Un système mixte de traitement de l'information neuronale destiné à la simulation des réseaux de neurones réalistes

Bilel BELHADJ MOHAMED, Jean TOMAS  
Université Bordeaux 1  
Laboratoire IMS – UMR 5218 CNRS  
351, cours de la Libération  
33405 TALENCE Cedex

**Email :** [bilel.belhadj@ims-bordeaux.fr](mailto:bilel.belhadj@ims-bordeaux.fr)

## Résumé

*Les secrets des tâches de perception, de contrôle et de classification, facilement réalisées par les organismes vivants, ne sont pas encore dévoilés. Les neurosciences combinent plusieurs domaines de recherche, tels que la biologie, les mathématiques et l'électronique dans le but de mieux comprendre l'activité neuronale dans le cerveau humain. La contribution de l'électronique se focalise, entre autres, sur l'émulation des systèmes neuronaux biologiques en partant des éléments de base de l'activité neuronale, à savoir les neurones. Ce papier présente une architecture mixte pour simuler des réseaux de neurones réalistes. Celle-ci fait appel à des modèles mathématiques proches de la biologie qui reproduisent fidèlement l'activité électrique des neurones vivants. Cette architecture est une réponse à la limitation des systèmes de calcul classique envers l'exécution des modèles de neurones complexes.*

## 1. Introduction

Doté d'un traitement heuristique et partiel de l'information neuronale, le fonctionnement du cerveau humain intéresse au plus haut point les scientifiques de multiples disciplines. On observe l'apparition de plusieurs types de systèmes qui essaient de réaliser des tâches de perception en se basant sur des modèles proches de la biologie [1][2]. Ces modèles, dits réalistes, sont en complexité croissante suite aux découvertes incessantes de nouvelles propriétés des systèmes neuronaux biologiques. Confrontés à des équations différentielles de plus en plus complexes ainsi que des dizaines de paramètres à manipuler, les systèmes de calcul classiques, tels que les processeurs, les microcontrôleurs et les DSPs, s'avèrent incapable d'offrir une solution d'implémentation des réseaux de neurones réalistes.

Les systèmes neuromorphiques sont une des solutions. Ce sont des systèmes spécifiques dédiés au calcul des fonctions neuronales. Ils peuvent intégrer les modèles de neurone et de synapse en plusieurs exemplaires sur une seule puce. La simulation de ces modèles se fait en temps réel (échelle biologique) ou même plus rapide [3]. La communication entre plusieurs puces nécessite le développement des méthodes d'assemblage et de routage

de l'information neuronale. Ce papier présente une architecture mixte d'un système neuromorphique dédié à la simulation des réseaux de neurones réalistes.

Pour simuler des réseaux de neurones réalistes, il faut garantir l'évolution en temps continu et en temps réel de l'évolution de l'état électrique de tous les neurones qui composent le réseau. Ces contraintes temporelles ne peuvent être atteintes qu'en utilisant des supports d'implémentation présentant des similitudes avec les systèmes biologiques ; c'est la raison pour laquelle nous avons choisi l'électronique analogique.

Les neurones communiquent entre eux en émettant des pulses électriques, ou encore potentiels d'action, qui, pour atteindre le(s) neurone(s) cible(s), passent par des structures intermédiaires de connexion, appelées synapses. L'activité de tout le réseau obéit à des règles de plasticité, appliquées au niveau des synapses, et qui sont responsables de l'aspect apprentissage dans les réseaux de neurones. Contrairement à l'évolution en temps continu du comportement d'un neurone, l'évolution des règles de la plasticité peut se faire en temps discret. En effet, la forme régulière des potentiels d'action peut être représentée par un type de donnée booléen, état haut (il y a un potentiel d'action) ou état bas (pas de potentiel d'action), d'où la tendance à implémenter numériquement la plasticité du réseau [3].

Une architecture de système mixte est alors proposée afin d'assembler des puces analogiques et d'assurer la communication entre eux d'une façon numérique. Le modèle de neurone utilisé ainsi que son implémentation fera l'objet de la section 2. Le modèle et les détails de l'implémentation de la plasticité sont résumés dans la section 3. La section 4 évoquera l'assemblage de ces composants pour constituer le système final.

## 2. Le modèle du neurone

Il existe plusieurs modèles de neurones dans la littérature. Les concepteurs de réseaux de neurones choisissent leur modèle selon les besoins de leur application. Pour la conception des réseaux de neurones réalistes, les modèles les plus adéquats sont ceux qui suivent le formalisme de Hodgkin-Huxley [4].

## 2.1 Le formalisme de Hodgkin-Huxley

Une cellule nerveuse est isolée du milieu extérieur par une membrane lipidique. Cette dernière comporte des canaux ioniques capables d'échanger certains types d'ions de part et d'autres de la membrane. Ces flux ioniques fluctuent le potentiel de la membrane, et c'est en dépassant un certain seuil de tension que le neurone génère un potentiel d'action (ou encore spike).

Le modèle de Hodgkin-Huxley utilise l'analogie entre condensateur/membrane de la cellule, conductance/canaux ioniques, et générateur de tension/potential d'équilibre pour reproduire ce phénomène biologique.

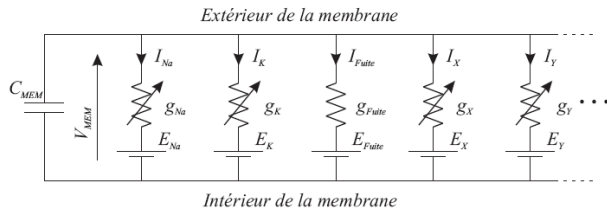


Figure 1. Le formalisme de Hodgkin-Huxley

La figure 1 montre la mise en circuit de tels modèles.  $C_{mem}$  représente la capacité de la membrane et  $V_{mem}$  son potentiel. Les canaux ioniques sont représentés par les conductances  $g_i$  où  $i \in \{Na, K, Fuite, x, y, \dots\}$ . Les canaux sodiques et potassiques sont les principaux facteurs des fluctuations de la membrane. Le canal de fuite représente les flux ioniques statiques de part et d'autres de la membrane. En plus de ces trois canaux de base, d'autres canaux supplémentaires à cinétique lente peuvent également être ajoutés au modèle. Ces canaux mettent en évidence des phénomènes d'accélération progressive (ou adaptation) et de ralentissement (ou modulation) de la fréquence des spikes. Par conséquent, plusieurs contributions de modèles peuvent être définies en ajoutant des canaux supplémentaires de notre choix. Cette constatation nous amène à considérer une évolution du modèle de Hodgkin-Huxley vers un formalisme de Hodgkin-Huxley.

L'expression des conductances  $g_i$  font la complexité de ce formalisme. Pour les principaux d'entre eux, la conductance prend la forme donnée par l'équation (1) :

$$g_i(V_{mem}) = G_i \cdot m^p(V_{mem}) \cdot h^q(V_{mem}) \quad (1)$$

Dans cette expression générique,  $G_i$  sont les conductances maximales que peut représenter chaque canal.  $m$  représente une activation (ouverture de canal) et  $h$  une inactivation (fermeture du canal).  $p$  et  $q$  sont deux entiers et représentent les forces des cinétiques. Selon leurs valeurs on détermine les équations des différents canaux ioniques. Pour les trois canaux principaux, ces valeurs valent :

$$\begin{cases} p = q = 0 & \text{canal de fuite} \\ p = 3, q = 1 & \text{canal sodium} \\ p = 4, q = 0 & \text{canal potassium} \end{cases}$$

Les grandeurs  $m$  et  $h$  répondent à la même équation différentielle représentée par l'équation (2).

$$\tau(V_{mem}) \cdot \frac{\partial x}{\partial t}(t) = x_{\infty}(V_{mem}) - x(t) \quad (2)$$

$$\text{Avec } x_{\infty}(V_{mem}) = \frac{1}{1 + \exp(s_f \frac{V_{mem} - V_{offset}}{V_{pente}})}$$

Le paramètre binaire  $s_f$  représente le signe de l'expression contenue dans l'exponentielle. Il dépend de la nature de la fonction modélisée :  $s_f = 1$  pour une inactivation, et  $s_f = -1$  pour une activation. Les paramètres  $V_{offset}$  et  $V_{pente}$  varient d'une conductance à une autre et spécifient la cinétique de cette dernière. Le  $\tau(V_{mem})$  est considéré dans notre cas comme une constante.

## 2.2 Implémentation sur ASIC analogique

L'implémentation des équations différentielles régies par les conductances des modèles du formalisme de Hodgkin-Huxley, a abouti à la réalisation d'une bibliothèque de fonctions analogiques. La bibliothèque a été réalisée en technologie *austriamicrosystems* BiCMOS 0,35  $\mu m$ . Cependant, les fonctions analogiques des conductances ne sont pas suffisantes pour modéliser un neurone. Des composants de configurations et de communication avec le milieu extérieur doivent aussi coexister. La figure 2 illustre l'architecture du circuit du neurone analogique réalisé au sein de notre groupe de recherche [5].

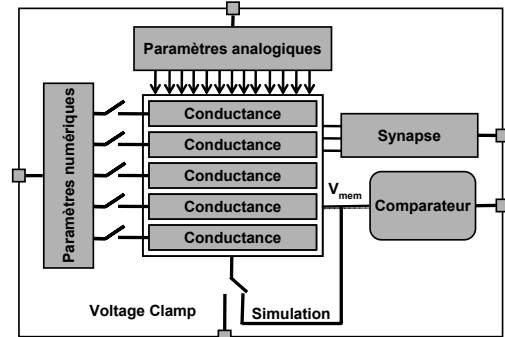


Figure 2. Architecture d'un neurone analogique

### 2.2.1 Configuration du neurone

Les modèles de neurones qui suivent le formalisme de Hodgkin-Huxley diffèrent par deux critères :

**Le nombre de conductances :** le modèle de neurone qu'on désire implémenter décide du nombre de conductances à connecter sur la capacité de membrane. Une fois décidé, des paramètres numériques seront envoyés pour ouvrir ou fermer les interrupteurs responsables de l'activation des conductances.

**Les paramètres des conductances :** il s'agit de régler les paramètres intrinsèques des conductances ( $V_{pente}$ ,  $V_{offset}$ ,  $p$ ,  $q$ ,  $\tau, \dots$ ). Le mode voltage clamp permet de déterminer les valeurs adéquates pour chaque conductance. Ces paramètres sont codés sous forme de tension et stockés dans des mémoires analogiques [6].

### 2.2.2 Communication avec l'extérieur

Le neurone ainsi configuré doit communiquer avec les autres neurones du réseau par le biais d'échange de signaux numériques. Deux composants réalisant l'interfaçage analogique/numérique sont ajoutés au circuit du neurone:

**La sortie du comparateur :** lorsque  $V_{mem}$  dépasse un certain seuil de tension, le composant comparateur transforme le signal analogique du potentiel de la membrane en un signal numérique et la redirige vers la sortie de l'ASIC.

**L'entrée synaptique :** le composant synapse reçoit une impulsion numérique dont la largeur code la valeur du poids synaptique. Comme illustré dans la figure 3, l'entrée synaptique commande une tension en rampe lorsqu'elle est à l'état haut (phase A), et autorise ensuite une évolution en exponentielle décroissante lorsqu'elle est à l'état bas (phase B). Le courant ainsi généré est appliqué sur la capacité de membrane.

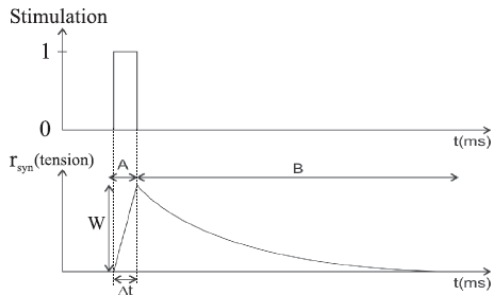


Figure 3. Fonctionnement d'une entrée synaptique

## 3. Modèle de la plasticité

La plasticité du réseau fait intervenir les instants d'apparition des spikes ainsi que leur fréquence pour déterminer la force de signal qui arrive au neurone cible. La plasticité entre deux neurones met en évidence une grandeur, appelée poids synaptique, dépendant de la fréquence des potentiels d'actions. Une connexion liant deux neurones est dite plastique, si son poids est variable. Dans le cas contraire, elle dite non plastique. Plusieurs algorithmes ont été définis pour simuler le changement du poids synaptique [3]. Cependant, l'algorithme de plasticité STDP (pour Spike-Timing Dependent Plasticity en anglais) est le plus adopté dans les neurosciences computationnelles grâce à sa capacité à reproduire des phénomènes d'apprentissage observés en biologie.

### 3.1 Spike-Timing Dependent Plasticity

Le modèle STDP, adopté dans ce travail, associe plusieurs types de dépendances de temps au trafic d'une paire de neurones [7]. L'équation (3) résume les principales caractéristiques du modèle en formulant la dérivée par rapport au temps du poids synaptique  $W_{ij}$  mis en jeu par les neurones  $N_i$  et  $N_j$ .  $P$  et  $Q$  sont des fonctions « mémoires » qui déterminent l'amplitude et le signe du changement de la plasticité et encode le temps et l'amplitude de l'influence mutuelle entre les spikes des neurones  $N_i$  et  $N_j$ . Les termes  $\epsilon$ , appelés aussi efficacité du potentiel d'action, indique que le

changement du poids synaptique est moins important lors de la dernière occurrence du potentiel d'action que lors des occurrences précédentes. Finalement, dans le but d'éviter une augmentation sans fin du poids synaptique, deux termes de saturation ont été ajoutés ;  $W_{LTP}$  et  $W_{LTD}$  avec  $W_{LTD} < W_{ij} < W_{LTP}$ .

$$\frac{dw_{ij}}{dt} = \epsilon_i \epsilon_j \left\{ (w_{LTP} - w_{ij}) \sum_{t_i} P[(t - t_i^{last}(t))] \delta(t - t_i) - (w_{ij} - w_{LTD}) \sum_{t_j} Q[(t - t_j^{last}(t))] \delta(t - t_j) \right\} \quad (3)$$

Avec

$$P(t) = \exp(-t / \tau_p), \quad \epsilon_i = 1 - \exp(-(t_i^{last} - t_i^{last-1}) / \tau_{post}),$$

$$Q(t) = \exp(-t / \tau_q), \quad \epsilon_j = 1 - \exp(-(t_j^{last} - t_j^{last-1}) / \tau_{pre}).$$

D'un point de vue pratique, les fonctions  $P(t)$  et  $Q(t)$  décrivent respectivement la potentiation à long terme (LTP) et la dépression à long terme (LTD) de la valeur du poids synaptique  $W_{ij}$ . La figure 4 illustre l'évolution de ses faits dans le temps.

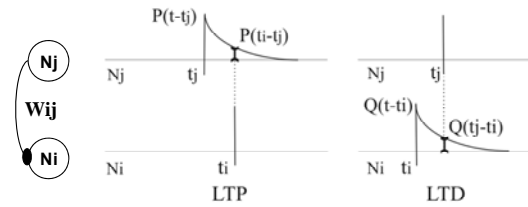


Figure 4. Illustration pratique de l'évolution de la STDP

## 3.2 Implémentation de la STDP sur FPGA

Lorsqu'un spike est généré par un neurone analogique, la sortie du comparateur est mise en état binaire haut. Cette information servira pour déclencher le calcul de la plasticité pour chaque paire de neurones. Si le nombre de paires de neurones mettant en jeu des connexions plastiques est important (de l'ordre des milliers), l'implémentation de la plasticité du réseau devient complexe et nécessite des techniques d'optimisation de calcul pour arriver à la faire tourner sur FPGA. Dans ce travail, on a utilisé la technique de multiplexage temporel pour le calcul de tous les poids synaptiques.

### 3.2.1 Le block STDP élémentaire

Le bloc STDP élémentaire implémente sur FPGA l'équation (3) pour une seule paire de neurones. Le schéma bloc de l'implémentation élémentaire de la STDP est fourni dans la figure 5. Les spikes générés par les neurones  $N_i$  et  $N_j$  sélectionnent la potentiation  $P$  ou la dépression  $Q$  du bloc de calcul de l'exponentielle décroissante. Les valeurs des  $\epsilon$  sont également extraites lors de l'arrivée des spikes. Ensuite les multiplications présentées dans l'équation (3) sont réalisées par l'intermédiaire d'un seul multiplieur pipeliné. Le résultat de sortie est la variation du poids synaptique  $\Delta W_{ij}$  qui est additionné ou soustrait à la valeur précédente du poids.

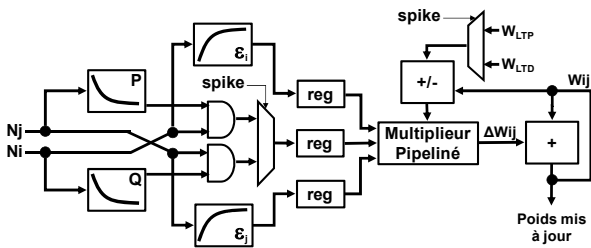


Figure 5. Schéma bloc de la STDP élémentaire

### 3.2.2 Implémentation de la STDP pour le réseau

Une première implémentation possible de la plasticité de tout le réseau est de dupliquer le bloc STDP élémentaire au nombre des connexions plastiques existantes dans le réseau. Cette solution, bien qu'elle soit la plus simple, ne permet pas d'implémenter tout le réseau sur FPGA. En effet, le nombre de connexions plastiques n'est pas fixe et risque de changer d'une configuration de topologie du réseau à une autre. En plus, le nombre de connexions plastiques évolue de façon quadratique par rapport au nombre de neurones mis en jeu par le réseau qui peut atteindre la centaine.

La deuxième solution repose sur le fait de sérialiser les opérations élémentaires du calcul de la plasticité, de manière à ne traiter que les connexions plastiques configurées par l'utilisateur. La technique utilisée sera de multiplexer dans le temps le fonctionnement d'un seul bloc STDP élémentaire. La figure 6 est une représentation simplifiée de cette implémentation.

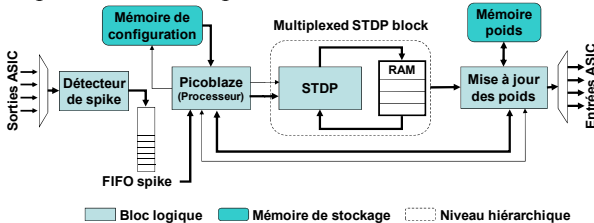


Figure 6. Schéma bloc de l'implémentation optimisée de la STDP

Le bloc STDP multiplexé est constitué d'un bloc STDP élémentaire et d'un bloc RAM. Cette architecture permet de calculer alternativement la plasticité de plusieurs connexions. Les valeurs des exponentielles décroissantes de la potentiation, la dépression et les  $\epsilon$  sont calculées et stockées dans une adresse mémoire spécifique et seront réutilisées par la suite pour recalculer les nouvelles valeurs.

La configuration de la topologie du réseau spécifié par l'utilisateur, est stockée dans une mémoire. Elle fournit des informations sur la connectivité et la plasticité du réseau. Un picoblaze utilise ces informations pour ordonnancer le calcul de la plasticité du réseau. Il alloue le bloc STDP multiplexé durant un slot de temps  $n$  pour calculer la variation du poids synaptique  $\Delta W_{ij}$  de la  $n^{\text{ième}}$  connexion plastique pour la stockée enfin dans la  $n^{\text{ième}}$  adresse de la RAM. La nouvelle valeur de l'ancien poids synaptique mis à jour est sauvegardée dans la mémoire.

## 4. Assemblage du système global

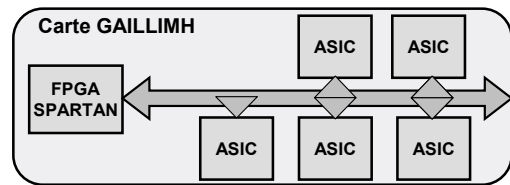


Figure 7. Architecture de la carte fille Gaillimh

L'assemblage de tous les composants du réseau se fait sur une carte électronique multicouche. Comme le montre la figure 7, cinq ASICs sont embarqués sur la carte. Chaque ASIC incorpore cinq neurones analogiques, soit un total de 25. La configuration des neurones ainsi que la plasticité de tout le réseau sont assurées par un FPGA. L'architecture obtenue constitue la carte fille Gaillimh. Une interface série lie la carte Gaillimh à un PC où on peut fixer les paramètres de configuration des neurones et de tout le réseau. Cette architecture peut être étendue en termes de taille de réseau. En effet, la combinaison de plusieurs cartes fille Gaillimh dans un rack permet d'atteindre la centaine de neurones analogiques. Cette extension fera l'objet d'un travail futur.

## 5. Conclusions

Les réseaux de neurones réalistes mettent en jeu des modèles de neurone et de plasticité du réseau proche des modèles biologiques. Les systèmes de calcul classiques s'inclinent devant la complexité de tels modèles. Le système présenté dans ce papier, est une tentative de faire tourner en temps réel un réseau de neurones réalistes. Les neurones suivent le modèle de Hodgkin-Huxley et sont implémentés sur des ASICs analogiques. Le calcul de la plasticité du réseau est assuré par un FPGA. L'assemblage de ces composants a abouti à un système mixte capable de gérer 25 neurones et 625 connexions plastiques en temps réel (temps biologique).

## Références

- [1] D. Roggen, S. Hofmann, Y. Thoma et D. Floreano, *Hardware Spiking Neural Network with Run-Time Reconfigurable Connectivity in an Autonomous Robot*, in NASA/DOD Conference on Evolvable Hardware, 2003.
- [2] T. Delbruck et S. C. Liu. *A silicon early visual system as a model animal*. Vision research, 44:17, pp 2083-2089, 2004.
- [3] G. Cauwenberghs, "Neuromorphic Learning VLSI Systems: A Survey", Neuromorphic systems engineering, pp 381-408, Kluwer Academy Publishers, 1998.
- [4] A. L. Hodgkin, A. F. Huxley et B. Katz. *Ionic currents underlying activity in the giant axon of the squid*. Arch. Sci. Physiology, 3, pages 129-150, 1949.
- [5] S. Saighi, J. Tomas, Y. Bornat, et S. Renaud. *A Conductance-based Silicon Neuron with Dynamically Tunable Model Parameters*. In 2<sup>nd</sup> Int. IEEE conf. on Neural Engineering, 285-288, 2005.
- [6] L. Buhry et Sylvain Saighi. *Réglage de paramètres neuronaux par des techniques de "voltage clamp" sur des ICs neuromimétiques*. Conf. nat. JNRDM 2008.
- [7] A. Destexhe, Z. F. Mainen, "Plasticity in Single Neuron and Circuit Computation", Nature review Neuroscience, vol. 6, pp. 789-795, 2004.