



Comprendre le monde,
construire l'avenir®



UNIVERSITE PARIS-SUD

ÉCOLE DOCTORALE : STITS

Laboratoire Fiabilisation des Systèmes Embarqués au CEA LIST

DISCIPLINE PHYSIQUE

THÈSE DE DOCTORAT

soutenue le 14/11/2012

par

Olivier BICHLER

Contribution à la conception d'architecture de
calcul auto-adaptative intégrant des nano-
composants neuromorphiques et applications
potentielles

Directeur de thèse :

Co-directeur de thèse :

Jean-Philippe BOURGOIN

Simon THORPE

Directeur de la stratégie et des programmes au CEA
DR au CNRS, Université Toulouse III Paul Sabatier

Composition du jury :

Président du jury :

Rapporteurs :

Examineurs :

Sylvie RENAUD

Lionel TORRES

Angelo ARLEO

Christian GAMRAT

Yusuf LEBLEBICI

Jacques-Olivier KLEIN

Professeur à l'Université de Bordeaux 1

Professeur à l'Université Montpellier 2

DR au CNRS, Université Pierre et Marie Curie

Ingénieur Chercheur au CEA LIST

Professeur à l'École Polytechnique Fédérale de Lausanne

Professeur à l'Université Paris-Sud 11

Version du 7 novembre 2012 (09:26).

Table des matières

Résumé	xi
Abstract	xiii
Remerciements	xv
Introduction	1
1 Traitement de l'information : apport des dispositifs memristifs et mise en perspective avec les neurosciences	7
1.1 Introduction	7
1.2 Les dispositifs memristifs	8
1.2.1 Caractéristiques statiques	9
1.2.2 Caractéristiques dynamiques	9
1.2.3 Intégration	11
1.2.4 Propositions architecturales	14
1.3 Traitement des données naturelles	16
1.3.1 Qu'entend-on par «donnée naturelle»?	16
1.3.2 Architectures classiques	18
1.3.3 Apprentissage versus programmation	19
1.4 Ce que nous apprennent les neurosciences	20
1.4.1 Codage de l'information	20
1.4.2 Mécanismes d'apprentissage	22
1.4.3 Architectures biologiques	23
1.5 Discussion et perspectives	25
2 Technologies mémoires émergentes et memristivité	27
2.1 Introduction	27
2.2 Nanoparticle-Organic Memory transistor (NOMFET)	28
2.2.1 Description	28
2.2.2 Modèle fonctionnel	30
2.2.3 Extraction des paramètres du modèle	31
2.2.4 Implémentation et simulations	34
2.2.5 Dépression synaptique et contrôle du gain	36
2.3 Mémoire à changement de phase (PCM, PRAM ou PCRAM)	38
2.3.1 Description	38
2.3.2 Caractérisation expérimentale	40
2.3.3 Modélisation	40
2.4 Conductive-Bridging RAM (CBRAM)	46
2.4.1 Description	47
2.4.2 Caractérisation expérimentale	48
2.5 Resistive RAM (RRAM ou ReRAM)	49

2.5.1	Description	49
2.5.2	Exemples de caractéristiques	51
2.6	Dispositifs à jonction tunnel	52
2.7	Discussion et perspectives	53
3	Outils de modélisation et de simulation	55
3.1	Introduction	55
3.2	Simulateur évènementiel Xnet	56
3.2.1	Justification et objectifs	56
3.2.2	Principe de fonctionnement	56
3.2.3	Études de cas	61
3.2.4	Comparaison avec les simulateurs existants	64
3.3	Modélisation et simulation au niveau circuit	66
3.3.1	Présentation des langages de description matériel utilisés	67
3.3.2	Exemple de co-simulation Python/Cadence	68
3.4	Discussion et perspectives	70
4	Étude d’architectures neuro-inspirées nouvelle génération pour le traitement de données naturelles	73
4.1	Introduction	73
4.2	Extraction non supervisée de motifs visuels	74
4.2.1	Méthodologie	75
4.2.2	Principe de l’apprentissage	77
4.2.3	Apprentissage de motifs à partir de capteurs de vision dynamiques	81
4.2.4	Robustesse et immunité au bruit	89
4.3	Détection de motifs auditifs dans du bruit	90
4.3.1	Méthodologie	91
4.3.2	Résultats des simulations	94
4.4	Apprentissage et classification de motifs	97
4.4.1	Méthodologie	97
4.4.2	Performances comparées avec un Perceptron	102
4.5	Discussion et perspectives	104
5	Implémentations matérielles sur des technologies memristives	105
5.1	Introduction	105
5.2	État de l’art	106
5.3	Démonstration d’une mémoire associative sur NOMFET	109
5.3.1	Introduction	109
5.3.2	Caractérisation du NOMFET	110
5.3.3	Méthodologie	111
5.3.4	Résultats expérimentaux	115
5.3.5	STDP asymétrique	119
5.4	Implémentation de STDP sur PCM	120
5.4.1	État de l’art	121
5.4.2	Synapse et architecture proposées	123
5.4.3	Résultats de simulation	128
5.5	Apprentissage STDP stochastique sur CBRAM	133
5.5.1	Synapse et architecture proposées	134
5.5.2	Résultats de simulation	137
5.6	Discussion et perspectives	140

6 Conclusion, réflexions et perspectives	143
6.1 Synthèse	143
6.2 Réflexions et questions ouvertes	144
6.2.1 Les grands projets en cours	144
6.2.2 Quelles limitations pour STDP?	147
6.2.3 Et l'implémentation des neurones?	148
6.3 Perspectives	149
Glossaire	151
Publications personnelles	153
Bibliographie	155
Index	171

Table des figures

1	Prévisions ITRS 2011, jusqu'à l'horizon 2026	1
1.1	Caractéristiques statiques des dispositifs memristifs	9
1.2	Principe de l'intégration de structures memristives dans un «crossbar» .	11
1.3	«Crossbar» de dispositifs memristifs de l'Université du Michigan	12
1.4	Illustration du problème des courants de fuite dans les «crossbar»	13
1.5	Principe des Complementary Resistive Switches	14
1.6	Circuits hybrides nano/CMOS (CMOL et FPNI)	16
1.7	Principe de la logique d'implication matérielle	17
1.8	Architecture de vision moderne pour la détection de piéton développée au DIASI	18
1.9	Détail de l'étape de détection/classification pour l'architecture de détec- tion de piéton du DIASI	19
1.10	Circuits impliqués dans la production des réponses des champs récep- teurs de la rétine	21
1.11	Réponse impulsionnelle des champs récepteurs de la rétine	22
1.12	STDP observée en biologie	23
1.13	Extraction non supervisée d'un motif répétitif dans du bruit	24
1.14	Principe d'extraction non supervisée de caractéristiques visuelles avec STDP	25
1.15	Caractéristiques visuelles extraites avec STDP	25
2.1	Structure physique du NOMFET	29
2.2	Configuration du NOMFET et circuit électrique équivalent pour la mo- délisation	29
2.3	Annotation du signal en entrée du NOMFET pour le modèle itératif . .	30
2.4	Courant en régime permanent du NOMFET : modèle vs mesures expé- rimentales	32
2.5	Comportement du NOMFET avec une séquence d'impulsions incluant différentes fréquences	33
2.6	Ajustement de courbe avec un NOMFET de largeur de canal 200 nm et des impulsions d'amplitude -3 V	33
2.7	Modélisation statistique des quatre paramètres physiques du modèle du NOMFET	34
2.8	Symbole du NOMFET et circuit de test dans l'éditeur de schéma de Cadence	35
2.9	Flot pour la co-simulation hybride nano/CMOS avec Matlab et Spectre de Cadence	35
2.10	Simulation en transitoire avec le modèle fonctionnel du NOMFET . . .	36
2.11	Circuit pour la détection d'un changement synchrone et non corrélé du taux d'activité en entrée de 10 NOMFETs	36

2.12	Simulation du détecteur de changement du taux d'activité basé sur des NOMFETs	37
2.13	Structure physique d'un dispositif PCM	38
2.14	Profils de température pour l'amorphisation et la cristallisation de la PCM	39
2.15	Caractéristique R-I des dispositifs GST et GeT	39
2.16	Caractéristiques expérimentales de LTP pour les dispositifs PCM GST d'IBM	40
2.17	Caractéristiques expérimentales de LTP pour les dispositifs PCM GST .	41
2.18	Caractéristiques expérimentales de LTPP pour les dispositifs PCM GeTe	41
2.19	Mesure expérimentale de la dérive de la résistance PCM dans le temps .	43
2.20	Simulation au niveau circuit de caractéristiques pour les dispositifs GST	44
2.21	Mécanisme de commutation bipolaire avec les électrolytes solides	47
2.22	Structure d'un dispositif CBRAM avec son transistor de sélection	48
2.23	Caractéristiques statiques et multi-niveaux de la CBRAM	48
2.24	Caractérisation de la stochasticité des dispositifs CBRAM	49
2.25	Mécanisme de commutation unipolaire pour les oxydes métalliques . . .	50
2.26	Mécanisme de commutation bipolaire pour les oxydes métalliques	51
2.27	Caractéristiques de la nano-synapse de l'Université du Michigan	51
2.28	Caractéristiques du memristor de HP Labs	52
2.29	Caractéristiques du memristor ferroélectrique de CNRS/Thales	52
2.30	Comparaison des différentes technologies pour une application synaptique	54
3.1	Traitement des évènements dans le simulateur Xnet	57
3.2	Structure de base pour un groupe de neurones dans Xnet	60
3.3	Stimuli d'entrée possibles dans Xnet	61
3.4	Principe de la modélisation d'une impulsion de durée non nulle dans Xnet	62
3.5	Données en sortie d'une simulation typique dans Xnet	63
3.6	Aperçu de la modélisation d'un système basé sur la PCM dans Xnet . .	64
3.7	Flot pour la simulation hybride nano-dispositif memristif/CMOS avec Python et Cadence	67
3.8	Méthodologie de conception d'une architecture en VHDL-AMS	68
3.9	Schéma de la «netlist» construite avec Python pour la co-simulation avec Cadence	69
3.10	Visualisation des signaux d'entrée et de la sortie du neurone pour le système d'apprentissage à base de PC	71
3.11	Évolution de la conductance des dispositifs PCM	71
4.1	Règle d'apprentissage STDP	76
4.2	Caractéristiques des trajectoires de balles utilisées comme stimuli d'entrée du réseau	77
4.3	Topologie du réseau de neurones pour l'extraction de trajectoires partielles	78
4.4	Évènements émis par les neurones de sortie en fonction du temps pour les trajectoires de balles	78
4.5	Activité des neurones après apprentissage pour les trajectoires de balles	79
4.6	Reconstruction des poids après apprentissage pour les trajectoires de balles	80
4.7	Illustration du jeu de données utilisé pour l'apprentissage de motifs complexes (trajectoire de véhicules)	81
4.8	Topologie du réseau de neurones pour l'apprentissage de trajectoires complexes	82
4.9	Reconstruction des poids synaptiques de la première couche après apprentissage de la séquence des véhicules	83
4.10	Détection des véhicules sur chaque voie de circulation après apprentissage	84

4.11	Détection des véhicules sur chaque voie de circulation après apprentissage (stratégie optimisée)	85
4.12	Reconstruction des poids pour la seconde couche après l'apprentissage couche par couche	86
4.13	Topologie du réseau de neurones pour l'apprentissage spatialement localisé	87
4.14	Reconstructions des poids pour quelques neurones champs récepteurs . .	88
4.15	Modélisation de la cochlée humaine	92
4.16	Mise en évidence de l'échelle ERB pour les filtres passe-bande du modèle de la cochlée	93
4.17	Construction d'un stimulus auditif pour l'extraction d'un motif répétitif	93
4.18	Trains d'impulsions générés par notre modèle de cochlée pour une tranche de bruit blanc	94
4.19	Apparition de sélectivité au motif de bruit blanc audio répétitif avec STDP	95
4.20	Évolution de la sélectivité du neurone durant l'apprentissage	95
4.21	Évolution de la sensibilité moyenne du neurone durant l'apprentissage pour 10 stimuli différents	96
4.22	Illustration de trois variations possibles d'un codage fréquentiel d'une image statique	97
4.23	Poids synaptique moyen \bar{w} en fonction de la probabilité de LTP	99
4.24	Cas limites du poids synaptique moyen \bar{w} en fonction de la probabilité de LTP	100
4.25	Stabilité des poids synaptiques pour différents modèles de synapses . . .	101
4.26	Illustration du calcul du taux de succès pour l'apprentissage non supervisé.	102
4.27	Reconstruction des poids appris dans une simulation avec 10 neurones (base MNIST)	103
4.28	Taux de reconnaissance durant l'apprentissage pour la base MNIST . .	103
5.1	Caractéristique dynamique d'un dispositif memristif bipolaire progressif et cumulatif	106
5.2	Propositions d'implémentation de STDP avec des dispositifs memristifs bipolaires	107
5.3	Notre proposition d'implémentation de la STDP simplifiée avec des dispositifs memristifs bipolaires	108
5.4	Caractéristique dynamique du NOMFET	111
5.5	Circuit électronique équivalent pour la mémoire associative	112
5.6	Forme des impulsions d'entrée et de rétroaction pour implémenter l'apprentissage associatif	113
5.7	Photographie de la carte expérimentale pour la mémoire associative NOMFETs	115
5.8	Schéma simplifié de la carte expérimentale pour un NOMFET	116
5.9	Dynamique de l'apprentissage associatif	117
5.10	Dynamique de l'apprentissage associatif avec oubli	117
5.11	Dynamique de l'apprentissage associatif sans rétroaction	118
5.12	Relation entre STDP symétrique et apprentissage associatif avec le NOMFET	119
5.13	Impulsions pré- et post-synaptiques pour l'apprentissage STDP avec le NOMFET	119
5.14	Fonction d'apprentissage STDP obtenue avec la carte d'interface électronique pour deux NOMFETs	120
5.15	État de l'art sur l'implémentation de STDP avec de la PCM	121
5.16	Règle STDP biologique et STDP simplifiée pour l'implémentation sur PCM proposée	122

5.17	Caractéristique de LTP expérimentale des dispositifs PCM et principe de la synapse 2-PCM	123
5.18	Opérations de lecture avec la PCM	124
5.19	Opérations d'écriture pour la règle STDP simplifiée avec la PCM	125
5.20	Principe du rafraichissement des synapses basées sur la PCM	126
5.21	Description des opérations de rafraichissement	126
5.22	Opérations de rafraichissement avec la PCM	127
5.23	Opérations de rafraichissement avec la PCM sans transistor de sélection	127
5.24	Topologie du réseau utilisé pour la simulation PCM	128
5.25	Carte de sensibilité des neurones de sortie et taux de détection avec la PCM	130
5.26	Variabilité simulée pour de la PCM GST et GeTe	131
5.27	Évolution des courants de RESET et de SET pour différentes technologies PCM	132
5.28	Proposition d'implémentation de STDP avec de la CBRAM	133
5.29	Caractéristiques de la modélisation binaire de la CBRAM	134
5.30	Règle d'apprentissage STDP stochastique	135
5.31	Principe du système de programmation stochastique	136
5.32	Principe de fonctionnement du générateur pseudo-aléatoire	136
5.33	Modes de fonctionnement : stochasticité extrinsèque, stochasticité intrinsèque et combinaison des deux	137
5.34	Détection des véhicules avec des synapses binaires	138
5.35	Apprentissage d'un motif répétitif caché dans du bruit audio	139
5.36	Évolution de la sensibilité du neurone durant l'apprentissage	139
5.37	Reconstruction des poids pour un apprentissage stochastique sur la base MNIST	140
5.38	Comparaison des performances entre des synapses analogiques et des synapses binaires	140
6.1	Vision du projet SyNAPSE de la DARPA	144
6.2	Vision du projet européen BrainScaleS	146

Liste des tableaux

2.1	Paramètres présents dans la fonction d'ajustement pour le NOMFET . .	32
2.2	Paramètres pour le modèle phénoménologique des courbes de LTP des dispositifs PCM	42
2.3	Paramètres pour les simulations avec le modèle semi-compact de la PCM GST	45
3.1	Paramètres de base d'une simulation Xnet	60
3.3	Performances comparées entre Xnet et Brian	66
4.1	Description et valeur des paramètres des neurones pour l'extraction de trajectoires partielles.	80
4.2	Moyenne et écart type pour les paramètres synaptiques	81
4.3	Paramètres des neurones pour l'apprentissage de motifs complexes . . .	85
4.4	Paramètres des neurones pour l'extraction du sous-motifs	87
4.5	Statistiques d'apprentissage, pour la durée complète de l'apprentissage .	88
4.6	Statistiques du taux de détection sur 100 simulations	89
4.7	Statistiques du taux de détection sur 100 simulations, avec dispersion sur les paramètres neuronaux	89
5.1	Paramètres des neurones pour l'apprentissage avec la PCM	129
5.2	Statistiques d'apprentissage, sur l'ensemble de la durée d'apprentissage avec la PCM	130
5.3	Statistiques énergétiques et consommation synaptique pour la PCM . .	132
5.4	Paramètres pour l'apprentissage stochastique de trajectoires de véhicules	138
5.5	Paramètres pour l'apprentissage stochastique de motifs auditifs	139

Résumé

Dans cette thèse, nous étudions les applications potentielles des nano-dispositifs mémoires émergents dans les architectures de calcul. Nous montrons que des architectures neuro-inspirées pourraient apporter l'efficacité et l'adaptabilité nécessaires à des applications de traitement et de classification complexes pour la perception visuelle et sonore. Cela, à un coût moindre en termes de consommation énergétique et de surface silicium que les architectures de type Von Neumann, grâce à une utilisation synaptique de ces nano-dispositifs.

Ces travaux se focalisent sur les dispositifs dit «memristifs», récemment (ré-)introduits avec la découverte du memristor en 2008 et leur utilisation comme synapse dans des réseaux de neurones impulsifs. Cela concerne la plupart des technologies mémoire émergentes : mémoire à changement de phase – «Phase-Change Memory» (PCM), «Conductive-Bridging RAM» (CBRAM), mémoire résistive – «Resistive RAM» (RRAM)... Ces dispositifs sont bien adaptés pour l'implémentation d'algorithmes d'apprentissage non supervisés issus des neurosciences, comme «Spike-Timing-Dependent Plasticity» (STDP), ne nécessitant que peu de circuit de contrôle.

L'intégration de dispositifs memristifs dans des matrices, ou «crossbar», pourrait en outre permettre d'atteindre l'énorme densité d'intégration nécessaire pour ce type d'implémentation (plusieurs milliers de synapses par neurone), qui reste hors de portée d'une technologie purement en «Complementary Metal Oxide Semiconductor» (CMOS).

C'est l'une des raisons majeures pour lesquelles les réseaux de neurones basés sur la technologie CMOS n'ont pas eu le succès escompté dans les années 1990. A cela s'ajoute la relative complexité et inefficacité de l'algorithme d'apprentissage de rétro-propagation du gradient, et ce malgré tous les aspects prometteurs des architectures neuro-inspirées, tels que l'adaptabilité et la tolérance aux fautes.

Dans ces travaux, nous proposons des modèles synaptiques de dispositifs memristifs et des méthodologies de simulation pour des architectures les exploitant. Des architectures neuro-inspirées de nouvelle génération sont introduites et simulées pour le traitement de données naturelles. Celles-ci tirent profit des caractéristiques synaptiques des nano-dispositifs memristifs, combinées avec les dernières avancées dans les neurosciences.

Nous proposons enfin des implémentations matérielles adaptées pour plusieurs types de dispositifs. Nous évaluons leur potentiel en termes d'intégration, d'efficacité énergétique et également leur tolérance à la variabilité et aux défauts inhérents à l'échelle nano-métrique de ces dispositifs.

Ce dernier point est d'une importance capitale, puisqu'il constitue aujourd'hui encore la principale difficulté pour l'intégration de ces technologies émergentes dans des mémoires numériques.

Mots clefs : memristor, dispositif memristif, réseau de neurones impulsifs, système neuromorphique, spike-timing-dependent plasticity.

Abstract

Title: *Adaptive Computing Architectures Based on Nano-fabricated Components.*

In this thesis, we study the potential applications of emerging memory nano-devices in computing architecture. More precisely, we show that neuro-inspired architectural paradigms could provide the efficiency and adaptability required in some complex image/audio processing and classification applications. This, at a much lower cost in terms of power consumption and silicon area than current Von Neumann-derived architectures, thanks to a synaptic-like usage of these memory nano-devices.

This work is focusing on memristive nano-devices, recently (re-)introduced by the discovery of the memristor in 2008 and their use as synapses in spiking neural network. In fact, this includes most of the emerging memory technologies: Phase-Change Memory (PCM), Conductive-Bridging RAM (CBRAM), Resistive RAM (RRAM)... These devices are particularly suitable for the implementation of natural unsupervised learning algorithms like Spike-Timing-Dependent Plasticity (STDP), requiring very little control circuitry.

The integration of memristive devices in crossbar array could provide the huge density required by this type of architecture (several thousand synapses per neuron), which is impossible to match with a CMOS-only implementation.

This can be seen as one of the main factors that hindered the rise of CMOS-based neural network computing architectures in the nineties, among the relative complexity and inefficiency of the back-propagation learning algorithm, despite all the promising aspects of such neuro-inspired architectures, like adaptability and fault-tolerance.

In this work, we propose synaptic models for memristive devices and simulation methodologies for architectural design exploiting them. Novel neuro-inspired architectures are introduced and simulated for natural data processing. They exploit the synaptic characteristics of memristives nano-devices, along with the latest progresses in neurosciences.

Finally, we propose hardware implementations for several device types. We assess their scalability and power efficiency potential, and their robustness to variability and faults, which are unavoidable at the nanometric scale of these devices.

This last point is of prime importance, as it constitutes today the main difficulty for the integration of these emerging technologies in digital memories.

Keywords: memristor, memristive device, spiking neural network, neuromorphic system, spike-timing-dependent plasticity.

Remerciements

Cette thèse a été effectuée d’abord au Laboratoire Calcul Embarqué (LCE), puis au Laboratoire Fiabilisation des Systèmes Embarqués (LFSE) du Département Architectures Conception & Logiciels Embarqués (DACLE) au Laboratoire d’Intégration des Systèmes et Technologies du Commissariat à l’énergie atomique et aux énergies alternatives (CEA) (CEA LIST), sur le plateau de Saclay. J’ai été accueilli au sein du Service Architectures et Conception (SARC) du Département Technologies et Systèmes Intelligents (DTSI) dans un premier temps, puis au sein du DACLE.

Je remercie Thierry Collette, ancien chef du SARC et chef du DACLE, ainsi que Jean-René Lèquepeys, son prédécesseur au DACLE et Denis Platter, ancien chef du DTSI, pour leur accueil et les moyens qu’ils m’ont fournis pour accomplir cette thèse dans les meilleures conditions.

Je tiens à remercier Raphaël David, chef du LCE, pour la confiance et le soutien qu’il m’a accordés tout au long de ces trois ans, ainsi que Laurent Letellier, qui m’a accueilli lorsqu’il était chef de ce laboratoire.

Je remercie également Fabrice Auzanneau et Yannick Bonhomme, respectivement ancien et nouveau chefs du LFSE pour leur accueil chaleureux dans leur laboratoire.

Je remercie Simon Thorpe, directeur de recherche au Centre national de la recherche scientifique (CNRS) et Jean-Philippe Bourgoïn, directeur de la stratégie et des programmes au CEA, pour avoir accepté de co-diriger cette thèse et pour avoir participé à mon jury. Je tiens à les remercier pour leur enthousiasme et leur soutien sans faille tout au long de ma thèse. Nos échanges réguliers m’ont permis de progresser efficacement dans la conduite de mes travaux et j’ai beaucoup appris de leurs conseils avisés.

Je remercie Christian Gamrat, ingénieur chercheur au CEA LIST, pour avoir effectué l’encadrement de ma thèse et pour avoir participé à mon jury. Je tiens à le remercier pour la confiance qu’il m’a accordée et pour la grande autonomie qu’il m’a laissée tout en étant toujours présent aux moments où j’en avais besoin.

Je remercie Lionel Torres, professeur à l’Université Montpellier 2, et Angelo Arleo, directeur de recherche au CNRS, pour avoir accepté de juger mon travail en tant que rapporteurs ainsi que pour leur participation à mon jury de thèse. Je remercie également Yusuf Leblebici, professeur à l’École Polytechnique Fédérale de Lausanne, Jacques-Olivier Klein, professeur à l’Université Paris-Sud 11, pour leur participation à mon jury et Sylvie Renaud, professeur à l’Université de Bordeaux 1, pour avoir accepté de présider mon jury de thèse.

Je tiens à remercier Weisheng Zhao et Damien Querlioz, avec qui j’ai collaboré lors-

qu'ils étaient en post-doc au CEA. Je leur dois beaucoup et sans eux cette thèse ne ressemblerait sans doute pas à ce qu'elle est aujourd'hui.

Je remercie mes collègues de l'Institut d'Électronique, de Microélectronique et de Nanotechnologie (IEMN), Fabien Alibert, Stéphane Pleutin et Dominique Vuillaume. C'est un réel plaisir de travailler avec eux et j'espère que cela pourra continuer dans le futur.

Je remercie mes collègues du Laboratoire d'électronique des technologies de l'information du CEA (CEA LETI), Barbara De Salvo, chef de l'Advanced Memory Technology Laboratory (AMTL), Ludovic Poupinet et Manan Suri, pour la confiance qu'ils m'ont accordée et l'excellent travail que nous avons accompli ensemble.

Je remercie Jean-Marc Philippe, ingénieur chercheur au CEA LIST, pour avoir répondu à ma demande de candidature spontanée au sein du laboratoire et m'avoir mis en relation avec Christian pour mon stage et ma thèse.

Je tiens à remercier l'ensemble des membres du LCE et du LFSE que j'ai pu côtoyer pendant ma thèse. La bonne ambiance de travail qui règne au sein du laboratoire a grandement contribué à la réussite de cette expérience. J'ai une pensée particulière pour les collègues - et amis - ayant partagé mon bureau.

Je remercie enfin l'ensemble de ma famille et de mes amis pour leur soutien et l'intérêt qu'ils ont porté à mes travaux.

Introduction

L'industrie et le monde de la recherche mettent aujourd'hui en œuvre des efforts considérables pour poursuivre l'augmentation exponentielle de la densité d'intégration, qui assure la croissance d'un grand nombre de secteurs économiques, alors que l'industrie de base du semi-conducteur pèse à elle seule près de 315 milliards de dollars annuels dans le monde en 2012¹. Celle-ci se heurte cependant de plus en plus aux limites physiques et des procédés de fabrication. Ainsi, la taille des transistors ne pourra vraisemblablement plus diminuer significativement en dessous de la dizaine de nanomètres, même si la loi de Moore devrait se poursuivre au moins jusqu'en 2025 notamment grâce à la lithographie ultraviolet extrême, aux transistors multi-grilles et à l'intégration 3D (voir figure 1). Pourtant, malgré l'augmentation exponentielle des capacités de calcul ces dernières décennies, les applications envisageables notamment dans les systèmes embarqués restent extrêmement éloignées des capacités cognitives humaines. Le traitement de données naturelles par exemple (vision, audition, ou plus généralement, perception), progresse lentement par rapport au rythme de l'évolution technologique. A cet égard, la reconnaissance de visages reste un problème de référence, compte tenu des applications envisageables, qui est toujours d'actualité (Zhao et al., 2003).

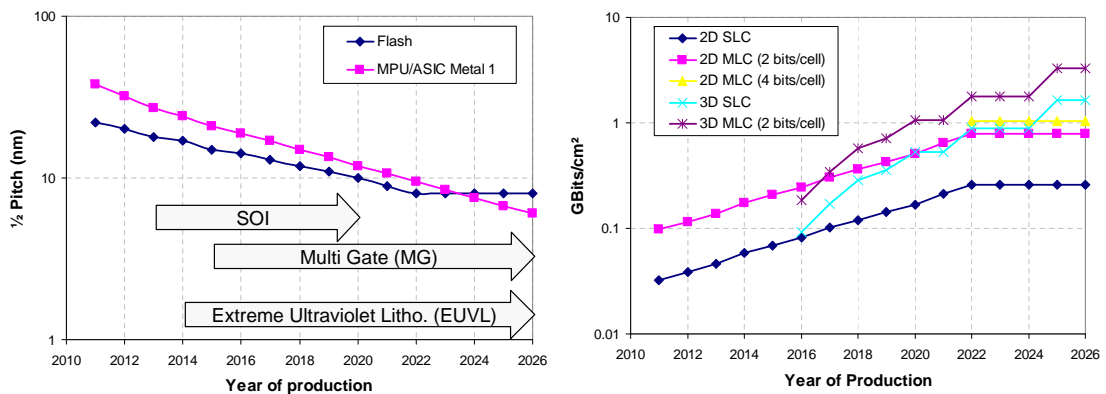


FIGURE 1 – Prévisions International Technology Roadmap for Semiconductors (ITRS) 2011 (ITRS, 2011), jusqu'à l'horizon 2026. A gauche, la diminution programmée du nœud technologique, pour la mémoire flash et la couche M1 pour les microprocesseurs/ASIC, ainsi que l'introduction des nouveaux procédés de fabrication devant permettre cette évolution. A droite, l'augmentation programmée de la densité de stockage pour les mémoires flash, qui devrait poursuivre une augmentation exponentielle grâce au «3D-stacking» et au «Multi-level Cell» (MLC).

S'il existe aujourd'hui des systèmes de reconnaissance très performants pour des problèmes contraints (cadrage, éclairage, ...), en reconnaissance de caractères et même de visages, ceux-ci saturant assez rapidement en performances dès que l'on relâche ces contraintes. En effet, il reste difficile de rendre de tels systèmes tolérants aux déformations, à commencer par les invariances en translation, rotation et à l'échelle. A cela

1. Source : International Data Corporation (IDC)

viennent généralement s'ajouter les variations liées aux conditions d'illumination et aux occlusions partielles. Dans ces systèmes, le gain en performances n'augmente malheureusement pas linéairement avec la quantité de mémoire et la puissance de calcul. Par exemple, sur la base de chiffres manuscrits MNIST (60 000 chiffres), pour passer d'un taux de reconnaissance de 95% à 99.7%, soit une amélioration relative de 5%, il faut multiplier la taille de la mémoire d'apprentissage par 50 (Querlioz et al., 2011a). Pour atteindre un taux de reconnaissance de 15.8% sur 20 000 catégories d'objets (800 par catégorie) sur la base ImageNet (Deng et al., 2009), 16 000 cœurs et 3 jours de calcul sont nécessaires pour faire évoluer un milliard de paramètres (Le et al., 2011), soit environ 100 fois plus de mémoire pour une amélioration relative de 70% par rapport au précédent état de l'art. Ce dernier exemple, l'un des systèmes les plus imposants en termes de quantité de mémoire utilisée pour l'apprentissage, représente très grossièrement l'équivalent d'environ 100 000 neurones biologiques (à raison de 10 000 synapses par neurone), soit une infime fraction du cerveau humain (0.0001%), alors que le cortex visuel en est le système le plus imposant.

De ces chiffres et de l'état de l'art, on peut retirer plusieurs choses. Tout d'abord, il est notable que les algorithmes de traitement d'image qui ont le plus de succès s'inspirent très fortement de l'organisation du système visuel dans le cerveau, qui est organisé en couches, en partant de la première couche V1, spécialisée dans la détection d'orientations et de contours. Malgré tout, la bonne manière de coder l'information dans un tel système reste une question ouverte, même si les progrès dans les neurosciences sont une source d'inspiration importante et légitime compte tenu du succès qu'ont les réseaux de neurones biologiques dans le traitement de l'information. Ensuite, les systèmes de reconnaissance les plus performants restent encore hors de portée des systèmes embarqués, dans lesquels la puissance de calcul et les ressources mémoires sont très contraintes et ce, même si l'apprentissage ou l'optimisation des paramètres sur des bases d'exemples se fait hors ligne. Pour des systèmes isolés ou temps réel, il n'est également pas forcément possible de déporter les calculs sur des serveurs distants, comme cela se fait déjà pour la reconnaissance vocale sur les smartphones. En outre, la puissance (énergétique) nécessaire pour faire fonctionner des algorithmes simples reste démesurée par rapport à l'énergie totale consommée par le cerveau, de l'ordre de la dizaine de watts (Kandel et al., 2000).

Plus fondamentalement, les raisons de ce décrochage entre l'évolution géométrique de la puissance de calcul brute des processeurs et les performances des systèmes de vision ou de perception sont à chercher dans la nature des opérations et des algorithmes en jeux. Alors que les opérations de traitement d'image sont typiquement locales et parallèles (convolutions, extractions de points d'intérêt, segmentation, reconnaissance), leur implémentation sur des processeurs de calcul reste de nature profondément séquentielle. Le «manycore» et les instructions «Single Instruction Multiple Data» (SIMD) ne modifient pas fondamentalement ce paradigme. Ces architectures continuent de se heurter au goulot d'étranglement («bottleneck») de Von Neumann (Backus, 1978), qui en isolant la mémoire du calcul induit un coût d'accès aux données et une limite sur le nombre de données pouvant être chargées en parallèle. Cette limite ne se résume toutefois pas à de simples choix architecturaux : elle est liée à la technologie sous-jacente. La plupart des architectures actuelles utilisent en effet extrêmement efficacement les transistors «Complementary Metal Oxide Semiconductor» (CMOS), en permettant d'effectuer n'importe quel type d'opération de manière programmable à l'aide d'une «Instruction Set Architecture» (ISA). Au contraire, les tentatives d'implémentation de réseaux de neurones artificiels en technologie CMOS, nombreuses dans les années 1990, n'ont pas connu le succès escompté. La difficulté à coupler la mémoire et le calcul (c'est-à-dire le poids synaptique et sa règle d'apprentissage associée) efficacement avec la technologie CMOS

ont toujours conduit à des systèmes qui deviennent rapidement obsolètes, devancés à la fois en puissance de calcul et en énergie par des implémentations logiciels sur «Central Processing Unit» (CPU)/«Graphics Processing Unit» (GPU), grâce à la loi de Moore. En particulier, il n'existe aucune solution satisfaisante pour lire et modifier simultanément des milliers d'éléments mémoires indépendamment les uns des autres et en fonction de leur état antérieur. L'implémentation de tels éléments mémoire peut nécessiter en CMOS plusieurs dizaines de transistors (Ishii et al., 1992). L'histoire se répète, puisqu'il existe aujourd'hui une résurgence de projets de ce type, portés par une nouvelle génération de réseaux de neurones (Schemmel et al., 2010; Furber et al., 2006; Choudhary et al., 2012).

Aujourd'hui, le rapprochement entre les progrès récents dans les neurosciences et dans le domaine des dispositifs mémoire laissent entrevoir la possibilité de créer des architectures radicalement nouvelles pour la perception de données naturelles. En éliminant le «bottleneck» de Von Neumann et avec un codage temporel de l'information, leur efficacité énergétique pourrait se rapprocher de celle du cerveau. Sur des applications de reconnaissance simples, déjà existantes dans des systèmes embarqués, on envisage ici de gagner plusieurs ordres de grandeur sur la consommation du système, qui pourrait être divisée par 100, 1000, 10 000 ou plus. A puissance égale, un tel gain permettrait également de multiplier les capacités des systèmes envisageables. Montrer que de telles architectures sont réalisables est l'objectif de ces travaux.

Dans le domaine des neurosciences tout d'abord, les expériences montrent qu'un humain est capable de reconnaître un objet en moins de 200 ms (Thorpe et al., 1996). Le codage de l'information dans le cerveau est évènementiel : l'information se propage d'un neurone à l'autre à l'aide de potentiels d'action («spikes»). Avec des fréquences mesurées de ces potentiels d'action de l'ordre de 1 à 100 Hz et compte tenu de l'organisation hiérarchique du système visuel, une telle réactivité du cerveau exclu un codage purement fréquentiel de l'information, puisqu'un neurone ne peut guère se déclencher plus d'une ou deux fois en moyenne durant un temps aussi court. Cela suggère également que le traitement de l'information se fait largement à sens unique («feedforward») (Thorpe and Imbert, 1989; Thorpe et al., 1996). Il existe aujourd'hui des pistes sérieuses sur la manière de réaliser un tel codage à partir de données visuelles ou auditives, combiné à des règles d'apprentissage adaptées de type «Spike-Timing-Dependent Plasticity» (STDP), elles aussi mises en évidence dans les réseaux de neurones biologiques (Dan and Ming Poo, 2004). On parle ainsi de la troisième génération de réseaux de neurones (Maass, 1997), après le modèle historique du perceptron, suivi par les premiers modèles de réseaux de neurones impulsifs avec codage fréquentiel. Ces progrès dans la compréhension des systèmes visuels biologiques inspirent fortement le développement des architectures de vision actuelles. *Montrer que ces réflexions mènent à applications concrètes constitue le premier enjeu majeur de ces travaux.*

Ensuite, dans le domaine des dispositifs mémoire, ces dernières années ont vu l'émergence de nombreuses nouvelles technologies mémoire, dont le but est de poursuivre l'augmentation de la densité d'intégration qui est de plus en plus compliquée à maintenir avec la mémoire flash (voir figure 1). On citera par exemple la mémoire à changement de phase – «Phase-Change Memory» (PCM), la «Conductive-Bridging RAM» (CBRAM), la mémoire résistive – «Resistive RAM» (RRAM) ou encore la mémoire magnétique – «Magnetic RAM» (MRAM). Si le principe physique de fonctionnement de chaque technologie est différent, celles-ci entrent toutes dans la classe des dispositifs dit memristifs (Chua and Kang, 1976), c'est-à-dire des dispositifs dont la conductance entre deux terminaux est modulée en fonction de la tension appliquée à leurs bornes et de leur état précédent. Alors que le concept de memristivité n'est pas nouveau, le lien avec ces technologies mémoire n'a été véritablement popularisé que récemment

avec le memristor (basé sur du TiO_2 , de la famille des RRAM) de Hewlett-Packard (HP) Labs (Strukov et al., 2008). Certaines de ces technologies présentent des caractéristiques extrêmement intéressantes pour en faire des éléments mémoire distribués de type synaptique, avec notamment la possibilité de modifier directement l'état mémorisé relativement à l'état précédent sans cycle de lecture-écriture à l'origine du «bottleneck» de Von Neumann. Cette caractéristique, leur compatibilité avec le CMOS, ainsi que la possibilité de lire et programmer ces dispositifs avec des courtes impulsions de tension, ouvre la voie à des implémentations efficaces des architectures évoquées ci-dessus. *Proposer des implémentations basées sur ces technologies permettant d'implémenter efficacement ces architectures constitue le second enjeu de ces travaux.*

Ces technologies mémoire émergentes peinent encore à concurrencer la technologie CMOS. Le principe de formation d'un filament conducteur dans un oxyde isolant, à la base de la RRAM, n'est pourtant pas nouveau (Hickmott, 1962), tout comme celui de la PCM. Une difficulté essentielle est le fait qu'à l'échelle nanométrique, les mécanismes de commutation en jeux ne sont plus entièrement déterministes. Cela a des conséquences importantes sur la variabilité des dispositifs et les défauts de programmation, qui augmentent avec la réduction de la taille des dispositifs. Or, le coût matériel pour corriger et éliminer ces défauts numériquement explose avec la probabilité de défaut. À noter que déjà, sur les mémoires flash avancées (en technologie CMOS classique), certains contrôleurs peuvent réserver jusqu'à 10% de la mémoire à cette fin. Cela limite également fortement le stockage multi-niveaux pour faire des mémoires numériques. Enfin, une intégration très dense, en matrice de composants, pose de nombreuses difficultés supplémentaires. Outre les défauts éventuels dans la matrice, la lecture et l'écriture d'un point mémoire ne doit pas perturber les dispositifs adjacents. *Montrer qu'une architecture qui utiliserait ces points mémoire comme des synapses est au contraire extrêmement tolérante à la fois à la variabilité des dispositifs et aux défauts, tout en permettant d'exploiter des matrices de composants efficacement constitue par conséquent le troisième enjeu majeur de ces travaux (Lee and Likharev, 2007) (Querlioz et al., 2011a).*

Organisation du mémoire Le mémoire est organisé de la manière suivante :

- Le premier chapitre, «Traitement de l'information : apport des dispositifs memristifs et mise en perspective avec les neurosciences», commence par introduire brièvement la notion de dispositif memristif. Il recense ensuite quelques uns des travaux les plus intéressants sur les propositions d'architectures basées sur ces dispositifs et examine de manière critique leur apport potentiel pour le traitement de l'information, leur faisabilité et leurs perspectives. Enfin, il revient plus en détail sur les idées développées dans cette introduction. Il dresse un état de l'état de l'art notamment dans le domaine de la vision, en ce qui concerne les architectures actuelles et le système visuel biologique.
- Le second chapitre, «Technologies mémoires émergentes et memristivité», détaille les différentes technologies memristives qui ont été considérées dans ces travaux, à savoir, le «Nanoparticle-Organic Memory Field Effect Transistor» (NOMFET), la PCM, la CBRAM et enfin la RRAM. Les dispositifs à jonction tunnel sont évoqués en raison de leur potentiel et de leur proximité (activités du laboratoire et d'équipes voisines). Les mécanismes physiques sont expliqués dans les grandes lignes et des modèles comportementaux sont proposés à partir des caractérisations expérimentales.
- Le chapitre suivant, «Outils de modélisation et de simulation», introduit les outils qui ont été développés pour mener à bien ces travaux.
- Le quatrième chapitre, «Étude d'architectures neuro-inspirées nouvelle génération pour le traitement de données naturelles», s'articule autour de trois applications

principales étudiées durant cette thèse : l'apprentissage et l'extraction de motifs visuels, l'extraction de motifs auditifs et la classification.

- Le cinquième chapitre, «Implémentations matérielles sur des technologies memristives», s'intéresse à l'implémentation des architectures présentées dans le chapitre précédent sur les différentes technologies memristives introduites dans le deuxième chapitre.
- Enfin, le chapitre de conclusion présente un ensemble de réflexions plus personnelles sur ces travaux et ma vision sur l'avenir de ce domaine, au travers de quelques questions qui me paraissent particulièrement pertinentes et importantes.

Chapitre 1

Traitement de l'information : apport des dispositifs memristifs et mise en perspective avec les neurosciences

Sommaire

1.1	Introduction	7
1.2	Les dispositifs memristifs	8
1.2.1	Caractéristiques statiques	9
1.2.2	Caractéristiques dynamiques	9
1.2.3	Intégration	11
1.2.4	Propositions architecturales	14
1.3	Traitement des données naturelles	16
1.3.1	Qu'entend-on par «donnée naturelle»?	16
1.3.2	Architectures classiques	18
1.3.3	Apprentissage versus programmation	19
1.4	Ce que nous apprennent les neurosciences	20
1.4.1	Codage de l'information	20
1.4.2	Mécanismes d'apprentissage	22
1.4.3	Architectures biologiques	23
1.5	Discussion et perspectives	25

1.1 Introduction

Dans ce chapitre, nous introduisons tout d'abord la notion de dispositif memristif, qui dans le contexte des technologies qui seront introduites dans le chapitre suivant se restreint généralement à celle de résistance programmable. Cette dénomination a été popularisée par le memristor de Hewlett-Packard (HP) Labs (Strukov et al., 2008). Elle est depuis couramment utilisée dans les publications architectures et circuits car elle permet de désigner formellement une famille de dispositifs indépendamment de la technologie. Nous présentons les propositions architecturales essentielles basées sur ces dispositifs, ainsi que les principales difficultés qu'elles posent, qui sont importantes pour implémenter de la mémoire et des circuits logiques.

Nous revenons ensuite en détail sur les problématiques évoquées dans l'introduction. Une première réflexion est menée sur le traitement des données naturelles, ses particularités et les difficultés que cela implique en termes d'implémentation et d'efficacité

calculatoire et énergétique. Elle est ensuite mise en perspective avec ce que nous apprennent les neurosciences, notamment dans le domaine de la vision et par conséquent sur la pertinence des démarches adoptées de ce point de vue. L'objectif est de souligner l'adéquation des dispositifs memristifs aux architectures pour la perception.

1.2 Les dispositifs memristifs

Un dispositif memristif est un composant électronique à deux terminaux qui se comporte comme une résistance dont la valeur change dynamiquement avec l'application d'un courant ou d'une tension à ses bornes. Ce changement dépend de l'état précédent du dispositif.

Avant de parler de dispositif memristif, Leon Chua a d'abord introduit le memristor (raccourci pour «memory resistor»), un composant passif théorique décrit comme le quatrième composant passif élémentaire, avec la résistance, le condensateur et l'inductance (Chua, 1971). Chua, en se basant sur un argument de symétrie et de complétude entre les équations liant les trois autres composants passifs élémentaires, a estimé qu'il manquait un quatrième composant liant le flux magnétique ϕ à la charge q . Mis en équation, cela donne

$$v(t) = M(q(t)).i(t) \text{ avec } \frac{dq}{dt} = i \quad (1.1)$$

i est le courant traversant le dispositif, v la tension à ses bornes et $M(q) = \frac{d\phi}{dq}$ est sa memristance. Chua a ensuite étendu sa théorie à toute une classe de systèmes, qu'il a appelé systèmes memristifs (Chua and Kang, 1976). Si l'on se restreint à des dispositifs à deux terminaux invariants dans le temps, la plupart des technologies mémoires peuvent se modéliser par les équations suivantes. Pour un dispositif contrôlé en courant :

$$v = R(x, i).i \quad (1.2)$$

$$\frac{dx}{dt} = f(x, i) \quad (1.3)$$

Et pour un dispositif contrôlé en tension :

$$i = G(x, v).v \quad (1.4)$$

$$\frac{dx}{dt} = h(x, v) \quad (1.5)$$

$x \in \mathbb{R}^n$ représente l'état interne (physique) du dispositif. Si une seule variable suffit à décrire l'état interne du dispositif, x est un réel. S'agissant d'un dispositif passif, le courant est toujours nul lorsque la tension est nulle et vice versa, ce quelque soit l'état x du système, qui intègre donc l'effet mémoire. Ces modèles ne pré-supposent pas du mécanisme physique, ni de la nécessité de l'existence d'un champ magnétique dans un dispositif réel.

Alors que l'invention du concept du memristor remonte à 1971, un modèle physique se comportant comme un memristor n'a été proposé qu'assez récemment par HP Labs avec son dispositif basé sur du TiO_2 (Strukov et al., 2008). Cette publication a suscité énormément d'intérêt dans le domaine, avec pas loin de 1000 citations depuis 2008, même si le modèle physique qui a été proposé a une plage de validité restreinte et ne se généralise pas à d'autres technologies dont les mécanismes de commutation sont

différents et complexes. La compréhension physique du memristor TiO_2 a d'ailleurs largement évolué dans les travaux qui ont suivi (Strukov et al., 2009; Pickett et al., 2009). Néanmoins, toutes les technologies de mémoire résistive sont modélisables par un modèle memristif (Chua, 2011). Ces technologies sont basées sur une structure métal-isolant-métal (MIM), dans laquelle un filament ou pont conducteur peut être créé ou détruit dans la couche isolante. Ces recherches ne sont pas nouvelles puisque les premières caractérisations de telles structures datent des années 1950 (Hickmott, 1962). Mais leur caractère memristif n'avait pas été identifié à l'époque et leur utilisation potentielle pour de la mémoire semblait assez éloignée.

1.2.1 Caractéristiques statiques

La caractéristique statique, ou quasi-statique, est obtenue avec un balayage lent de la tension appliquée aux bornes du dispositif entre les valeurs minimales et maximales admissibles (typiquement un signal triangulaire). Sur les dispositifs qui ont un effet mémoire, donc de type memristif, celle-ci fait apparaître le cycle d'hystérésis quasi-statique constitué des caractéristiques ON, ou «Low Resistance State» (LRS) (résistance minimale R_{on}) et OFF, ou «High Resistance State» (HRS) (résistance maximale R_{off}). Ces caractéristiques sont généralement linéaires autour de 0, ce qui signifie que la plupart des dispositifs non volatiles se comportent comme une simple résistance sous un faible champ électrique. Le passage de OFF à ON est communément appelé le SET tandis que le passage de ON à OFF est le RESET.

On distingue de manière générale deux types de caractéristiques statiques, selon que les passages de R_{off} à R_{on} et de R_{on} à R_{off} se font dans le même quadrant dans le plan I-V (unipolaire) ou non (bipolaire), comme illustré figure 1.1 (Waser and Aono, 2007). Le type de caractéristique dépend du mécanisme de commutation dans la structure MIM et diffère selon les matériaux utilisés. La mémoire à changement de phase – «Phase-Change Memory» (PCM) est unipolaire et la «Conductive-Bridging RAM» (CBRAM) est bipolaire, comme une majorité des dispositifs de type mémoire résistive – «Resistive RAM» (RRAM).

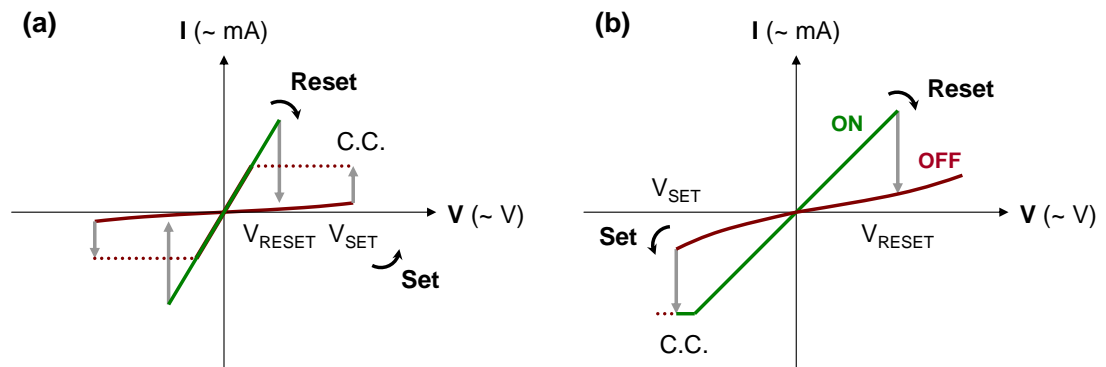


FIGURE 1.1 – Classification des caractéristiques statiques des dispositifs de type MIM. (a) Commutation unipolaire : les tensions de SET et de RESET sont toutes les deux du même signe. (b) Commutation bipolaire : le SET a lieu dans la polarité opposée au RESET. Il peut être nécessaire d'imposer une limitation en courant («current compliance») lors du SET pour éviter un claquage du dispositif.

1.2.2 Caractéristiques dynamiques

Dans le cadre d'une utilisation mémoire (synaptique ou numérique), les dispositifs memristifs ne sont pas utilisés en régime continu, mais en régime impulsionnel : la

lecture et la modification de l'état se font par des impulsions de tension. Plusieurs caractéristiques sont désirables pour un élément mémoire.

Multi-niveaux

Le stockage de plusieurs bits par élément mémoire est essentiel pour gagner en densité d'intégration et fait l'objet d'un effort de recherche conséquent. Les contraintes de ce point de vue ne sont pas les mêmes pour de la mémoire numérique ou pour une synapse. La mémoire numérique ne tolère pas d'erreur et nécessite des niveaux de résistances suffisamment stables et déterministes pour garantir une marge de bruit suffisante à un nombre de niveaux donné. Du stockage multi-niveaux allant de 2 à 4 bits par dispositif, soit 4 à 16 niveaux de résistance différents, a été montré expérimentalement pour la PCM (Nirschl et al., 2007; Papandreou et al., 2011) et la CBRAM (Kund et al., 2005; Schrogmeier et al., 2007), mais n'est pas pour le moment industrialisé. Dans les deux cas, le processus de programmation consiste en une boucle programmation-vérification, ce qui implique un circuit complexe susceptible d'être un facteur limitant important des performances de la mémoire en écriture. Ainsi, stocker de manière fiable 8 bits par dispositif, soit 256 niveaux, est un véritable défi, qui semble très difficile à relever d'autant plus que la variabilité et le caractère non déterministe des dispositifs tend à augmenter quand on diminue la taille des cellules. Une précision de programmation relative de 1% par rapport à l'objectif cible a été montrée récemment sur un dispositif memristif (Alibart et al., 2012a), ce qui équivaut à une précision de l'ordre de 7 bits. Au prix toutefois de plusieurs centaines d'étapes de programmation en moyenne pour atteindre le niveau désiré. Une synapse, elle, se comporte comme une mémoire analogique et n'exige pas de niveaux discrets bien définis, comme nous le verrons dans la suite de ces travaux.

Cumulativité

Par «cumulativité», nous entendons la possibilité de modifier la résistance du dispositif, dans un sens ou dans l'autre, relativement à son état précédent sans devoir le lire au préalable. Cette caractéristique, absente des mémoires classiques, est d'un immense intérêt pour une synapse. La cumulativité permet d'implémenter efficacement les algorithmes d'apprentissage synaptiques (qu'il s'agisse de «Spike-Timing-Dependent Plasticity» (STDP) ou de rétro-propagation du gradient par exemple) puisqu'elle permet d'augmenter ou de diminuer progressivement le poids synaptique relativement à sa valeur précédente sans nécessiter de cycle lecture-incrément/décrément-écriture. C'est précisément cette caractéristique qui permettrait de s'affranchir du «bottleneck» de Von Neumann qui rend complexe et inefficace les implémentations de réseaux de neurones sur des architectures «Complementary Metal Oxide Semiconductor» (CMOS) (Jin et al., 2010; Schemmel et al., 2006).

Stochasticité

Il s'agit ici de la probabilité qu'a le dispositif de commuter d'un état à l'autre selon les conditions de programmation utilisées. Un dispositif au comportement stochastique serait catastrophique dans une mémoire numérique, dans laquelle la probabilité de commuter doit être de 1 en toutes circonstances. Les impulsions de programmation pour la CBRAM doivent être suffisamment fortes (en amplitude et durée) pour garantir une commutation et ce avec une marge de sécurité suffisante, ce qui impose une limite basse de l'énergie nécessaire à la programmation. Concernant la famille des RRAM, une formation probabiliste du filament conducteur a été mise en évidence et caractérisée sur les dispositifs Ag/a-Si/p-Si de l'Université du Michigan (Jo et al., 2009b).

1.2.3 Intégration

Les technologies PCM, CBRAM et RRAM sont des structures de type MIM. Ces structures peuvent potentiellement être intégrées entre deux couches de métal. Il est ainsi possible de former une matrice de composants, ou «crossbar», pour obtenir la plus grande densité d'intégration possible, comme illustré sur la figure 1.2. En outre, la possibilité d'intégrer ces structures entre les couches de métal permettrait d'empiler des «crossbar» sur plusieurs niveaux au dessus du CMOS (comme pour les capacités MIM). En plus de multiplier la densité d'intégration, cela faciliterait la connectivité avec le CMOS, qui pourrait se faire verticalement. Cette perspective contribue à motiver un effort de recherche important dans ces technologies. Elle se heurte toutefois à de nombreux obstacles pour réaliser de la mémoire numérique comme nous le verrons dans la suite.

La taille des cellules formées à l'intersection de deux pistes de métal pourrait à terme avoisiner les 10×10 nm pour les trois technologies citées précédemment (ITRS, 2011). Des puces mémoires de grande capacité ont déjà été réalisées avec de la PCM et de la mémoire magnétique – «Magnetic RAM» (MRAM) : 1 Go en 58 nm par Samsung Electronics Co. Ltd. (Chung et al., 2011) et 64 Mo en 65 nm par Toshiba Corporation (Tsuchida et al., 2010) respectivement. Concernant la technologie RRAM, un «crossbar» de 32×32 lignes, soit 1024 cellules avec une structure Ag/a-Si/p-Si a été démontrée avec 92% de dispositifs fonctionnels (Jo et al., 2009a). Plus récemment, la même équipe a montré un «crossbar» entièrement fonctionnel de 40×40 avec un demi-pas de 50 nm (Kim et al., 2012). Contrairement aux puces mémoires précédentes, basées sur une architecture classique, ces «crossbar» peuvent être programmés sur plusieurs niveaux sans dispositif de sélection.

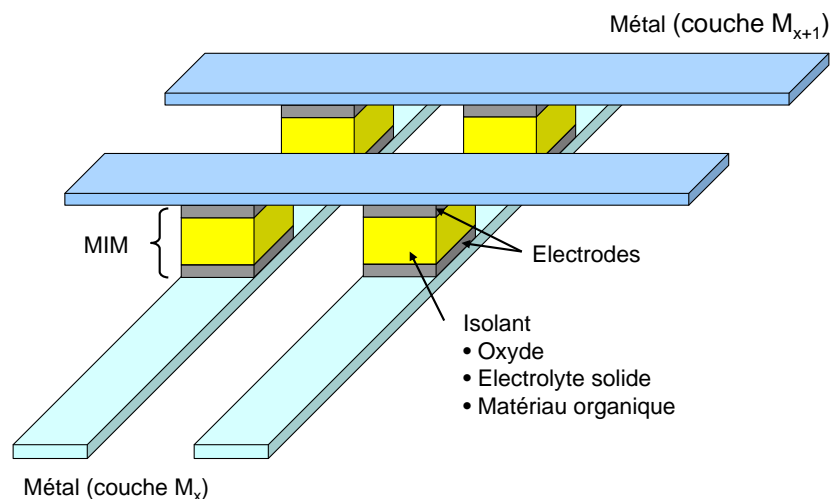


FIGURE 1.2 – Intégration de structures memristives de type MIM dans un «crossbar» entre deux couches de métal, sans dispositif de sélection, rendant possible une densité d'intégration extrêmement importante.

L'intégration de nano-dispositifs memristifs en «crossbar» pour réaliser de la mémoire ou de la logique se heurte toutefois principalement à deux points durs : les défauts, c'est-à-dire des dispositifs qui ne commutent pas et restent dans un état ON ou OFF et les courants de fuite, qui limitent fortement la taille des matrices passives réalisables en pratique.

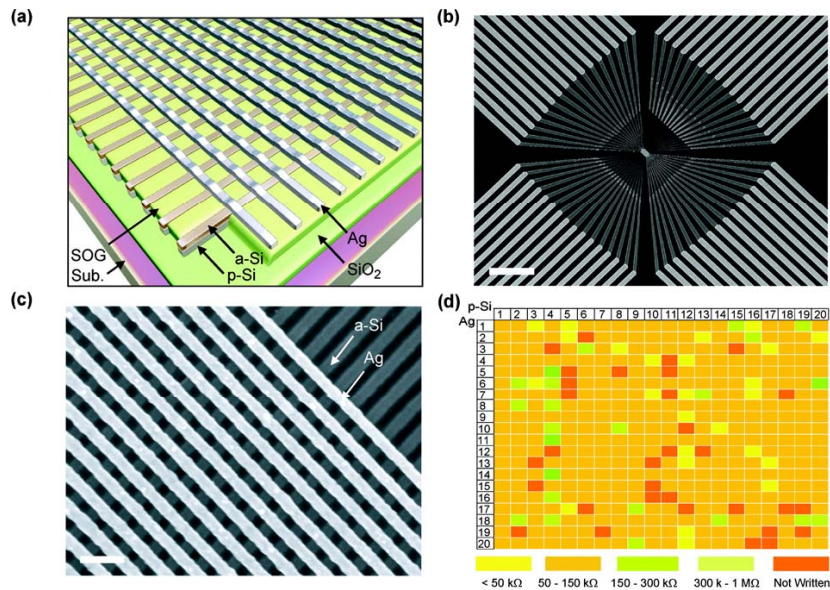


FIGURE 1.3 – «Crossbar» de l'Université du Michigan (Jo et al., 2009a). (a) «Crossbar» formé de nanofils constitués d'électrodes en Ag et p-Si et d'une couche active en a-Si. (b) Image obtenue par microscopie électronique à balayage – «Scanning Electron Microscopy» (SEM) d'une matrice de 1 Kb. Échelle : 40 μm . (c) Zoom de l'image précédente. (d) Carte des états de 400 points mémoire après envoi d'une impulsion de programmation. Certains n'ont pas commuté.

Dispositifs défectueux

Outre la variabilité des dispositifs, qui est un facteur limitant pour le stockage multi-niveaux, la gestion des dispositifs défectueux pose un problème supplémentaire dans une structure en «crossbar». Idéalement, afin de gagner en densité d'intégration, une matrice la plus grande possible est désirable. Mais cela rend plus difficile et plus coûteux l'isolement d'un dispositif défectueux, si cela implique par exemple d'éliminer la ligne d'adressage du «crossbar» contenant celui-ci. En outre, un taux de défaut de 1% signifie, dans tous les cas, en moyenne un défaut par ligne ou par colonne, ce qui est ingérable pour de la mémoire numérique. Des mécanismes de correction d'erreur sont possibles, mais ne sont plus efficaces passé un faible niveau de défaut, de 0.1 à 0.5% (Kuekes et al., 2005).

Ce problème est moins pénalisant pour de la logique reconfigurable. En effet, si la matrice de cellules mémoires est utilisée pour l'interconnexion de fonctions logiques, une stratégie possible est de d'abord lire l'ensemble de la matrice et de caractériser les défauts, pour pouvoir ensuite calculer un routage qui contourne les dispositifs défectueux. Un taux de circuits fonctionnels de 99% a été montré avec ce type d'approche pour 20% de dispositifs défectueux (Strukov and Likharev, 2005).

Courants de fuite

Un «crossbar», ou une matrice passive, c'est-à-dire sans dispositif de sélection comme ceux des figures 1.2 et 1.3, est essentiel pour atteindre les densités d'intégration promises. L'ajout d'un transistor ou d'une diode de sélection pour chaque dispositif est en effet une limitation sérieuse, car il n'existe pas pour le moment de technique d'intégration satisfaisante avec la structure MIM (ce qui exclut l'empilement 3D de matrices sur plusieurs niveaux de métal). Cependant, sans dispositif de sélection, les courants de fuite («sneak paths») limitent fortement la taille du «crossbar». Le pro-

blème se pose à la fois pour la lecture et pour l'écriture. Les courants parasites dans les dispositifs le long d'une ligne produisent une chute de la tension de programmation à l'extrémité de la ligne (voir figure 1.4). Il est à noter que même avec des sélecteurs, la résistance de la ligne de métal provoque une chute de tension aux bornes du dispositif sélectionné en formant un pont diviseur de tension avec celui-ci. En outre, ces chutes de tension dépendent de l'état des dispositifs memristifs et donc des données stockées dans le «crossbar». Minimiser ces courants parasites implique d'augmenter R_{on} et le ratio R_{off}/R_{on} (Liang and Wong, 2010). Pour atteindre une taille de 1000×1000 , R_{on} doit être de l'ordre du $M\Omega$, ce qui est supérieur à la plupart des dispositifs dont les caractéristiques sont publiées dans la littérature.

La nécessité d'avoir un R_{on} élevé peut toutefois avoir pour effet d'augmenter significativement les temps de propagation dans le «crossbar», du fait des capacités de ligne parasites. Cela pourrait limiter la vitesse de lecture dans le cas d'une mémoire numérique. Nous verrons dans la suite que cela n'est pas un facteur limitant dans le cas d'une utilisation synaptique, car les fréquences en jeu sont faibles.

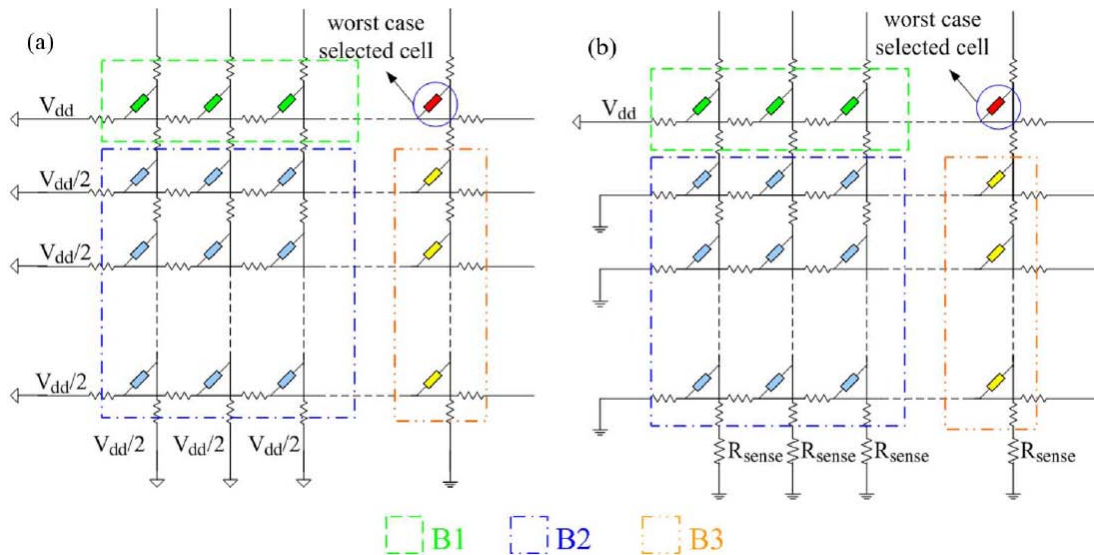


FIGURE 1.4 – Opération d'écriture (a) et de lecture (b) (Liang and Wong, 2010). La cellule pire-cas est celle la plus éloignée des sources de tension ligne/colonne. L'étude du pire cas peut se simplifier en considérant trois groupes de cellules, avec toutes les cellules dans le même état dans chaque groupe (B1, B2 et B3).

Un assemblage de deux dispositifs memristifs montés en tête-bêche pour former un «Complementary Resistive Switches» (CRS) a été proposé pour résoudre ce problème (Linn et al., 2010). Le principe est expliqué figure 1.5. Il exploite la caractéristique combinée de deux dispositifs pour toujours passer par un état de forte résistance au voisinage de 0 V. Cette stratégie résout en partie les problèmes d'intégration d'une diode ou d'un transistor aux cotés du dispositif memristif. Cependant elle pose des contraintes de fabrication supplémentaires et suppose des caractéristiques symétriques entre les deux dispositifs, ce qui rend le CRS peu robuste à la variabilité. En outre, une programmation multi-niveaux avec une telle structure n'est pas démontrée. L'introduction d'une non-linéarité directement dans la caractéristique du dispositif memristif pourrait également être une solution prometteuse (Yang et al., 2012).

Une utilisation en mémoire numérique nécessite une marge de bruit suffisante à la lecture d'un élément mémoire. Les courants de fuite ont pour effet de diminuer la différence ΔI entre les courants mesurés en bout de ligne pour les états R_{on} et R_{off} d'un dispositif dans la matrice. Cette différence doit être supérieure à la marge de bruit

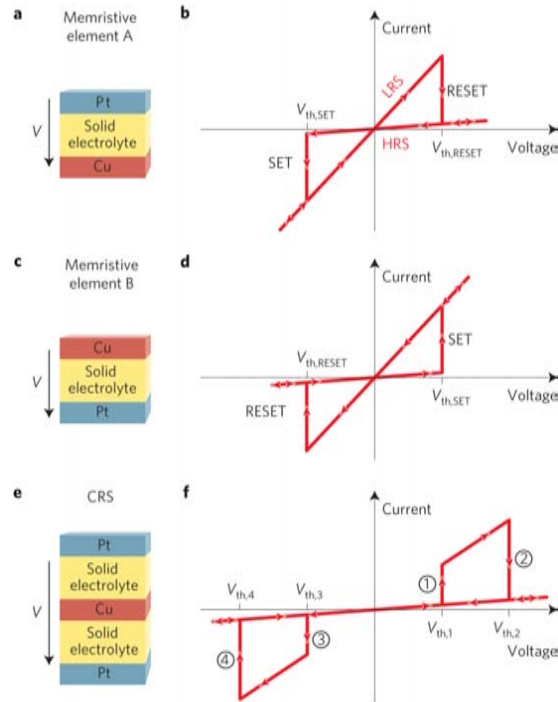


FIGURE 1.5 – Principe du CRS (Linn et al., 2010). (a) Dispositif memristif Pt/électrolyte solide/Cu (famille des CBRAMs). (b) Caractéristique I-V du dispositif. (c) Dispositif memristif Cu/électrolyte solide/Pt. (d) Caractéristique I-V du dispositif. (e) Les dispositifs memristifs sont assemblés en tête-bêche pour former un CRS. (f) Caractéristique de la pile CRS ainsi obtenue : quel que soit l'état du dispositif, on reste sur la caractéristique R_{off} au voisinage de 0 V.

de l'amplificateur d'instrumentation utilisé pour la lecture pour pouvoir distinguer les deux états (Liang and Wong, 2010). Cette contrainte est démultipliée pour l'exploitation dans le «crossbar» de dispositifs en multi-niveaux. Cette notion de marge de bruit est absente dans un réseau de neurones où les courants lus augmentent progressivement le potentiel de membrane du neurone jusqu'à le déclencher au-delà d'un seuil. L'effet des fuites de courant a dans ce cas globalement pour effet de pondérer les poids synaptiques selon l'emplacement des synapses dans la matrice. Le dispositif dans le pire cas de la figure 1.4 contribue en moyenne moins à l'intégration du neurone que tous les autres, quel que soit son état.

1.2.4 Propositions architecturales

Depuis la publication de HP Labs introduisant le memristor (Strukov et al., 2008), de nombreux travaux proposant d'exploiter ce type de dispositif dans des architectures numériques, analogiques ou neuromorphiques ont été publiés. Un bon nombre des approches proposées restent assez conceptuelles. Une proposition réaliste doit en effet traiter d'un grand nombre de problématiques :

- Le réalisme du modèle et de la simulation. Le modèle du memristor introduit dans (Strukov et al., 2008) n'est vérifié que sur la faible plage de tension pour laquelle la caractéristique du dispositif est linéaire. D'autres modèles «génériques» de dispositifs memristifs bipolaires y ajoutent une non-linéarité (Pershin et al., 2009; Zamarreño-Ramos et al., 2011). Enfin, les modèles plus précis s'appuient sur les données physiques d'une technologie et intègrent la réponse mesurée des dispositifs pour les conditions de programmation utilisées en simulation.
- La prise en compte des défauts. Pour les architectures logiques, cela peut inclure

- typiquement de la redondance, des codes correcteurs d'erreur ou un cycle de caractérisation/programmation post-fabrication.
- La prise en compte de la variabilité. Pour la programmation multi-niveaux (analogique ou numérique), ce point n'est traité de manière réaliste que par des boucles programmation-vérification.
 - Le passage à l'échelle. Le passage d'une brique élémentaire à base de dispositifs memristifs à un circuit complet et fonctionnel est complexe. La génération des signaux de contrôle et des impulsions de tension pour la programmation peut représenter un surcout très important et facilement ignoré dans les études théoriques.
 - L'intégration avec le CMOS. Peu de dispositifs expérimentaux sont fabriqués avec des procédés industriels standards. Cela rend incertain leur intégration à grande échelle et co-intégration avec le CMOS.

Interconnexions programmables

L'utilisation de dispositifs memristifs comme interconnexions programmables au dessus de portes logiques est l'idée la plus aboutie, avec la démonstration d'un circuit intégré (Borghetti et al., 2009; Xia et al., 2009). Cette idée a été introduite avec l'approche «CMOS/molecular» (CMOL) (Strukov and Likharev, 2005), qui consiste à interconnecter des inverseurs avec des dispositifs memristifs qui réalisent de la logique OU câblée. Une évolution ultérieure de HP Labs propose plutôt d'interconnecter des portes logiques pour réaliser un «Field-Programmable Nanowire Interconnect» (FPNI) (Snider and Williams, 2007). Les deux approches sont résumées figure 1.6. L'idée essentielle est d'intégrer des interconnexions memristives au dessus du CMOS avec un angle entre les lignes formant les matrices d'interconnexion et les portes CMOS. Cela permet de maximiser la densité d'intégration puisque l'intégralité des connexions entre la couche CMOS et les matrices memristives se fait avec des vias sans routage autour. Dans cette architecture, les interconnexions sont reprogrammées pour former un circuit logique à la manière d'un «Field-Programmable Gate Array» (FPGA) classique.

Autres circuits logiques

(Borghetti et al., 2010) a proposé un concept intéressant pour réaliser avec les mêmes dispositifs memristifs la fonction logique et la mémorisation, combinant ainsi dans un seul bloc élémentaire la porte logique et le registre. Il s'agit d'implémenter un bloc logique d'implication matériel, qui est l'un des éléments logique de base permettant de réaliser tout type de fonction logique. Le principe est illustré figure 1.7. Si l'idée est élégante, la génération des signaux de programmation et de contrôle reste problématique et aucun circuit à plus grande échelle basé sur cet élément n'a été présenté pour l'instant. Une bascule logique basée sur un memristor a également été montrée par la même équipe, avec un taux d'erreur de 0.1% sur la récupération de l'état stocké (Robinett et al., 2010), ce qui la laisse pour l'instant inexploitable.

Il a été également proposé d'utiliser de la mémoire memristive pour implémenter les cellules de mémoire adressable par contenu – «Content-Addressable Memory» (CAM), classiquement basées sur de la SRAM (Alibart et al., 2011).

Autres applications

Outre les applications neuromorphiques, dont l'état de l'art sera traité au cœur du chapitre «Implémentations matérielles sur des technologies memristives», un grand nombre d'autres applications ont été proposées autour du memristor ou de dispositifs memristifs. Celles-ci n'entrent pas directement dans la problématique de ces travaux.

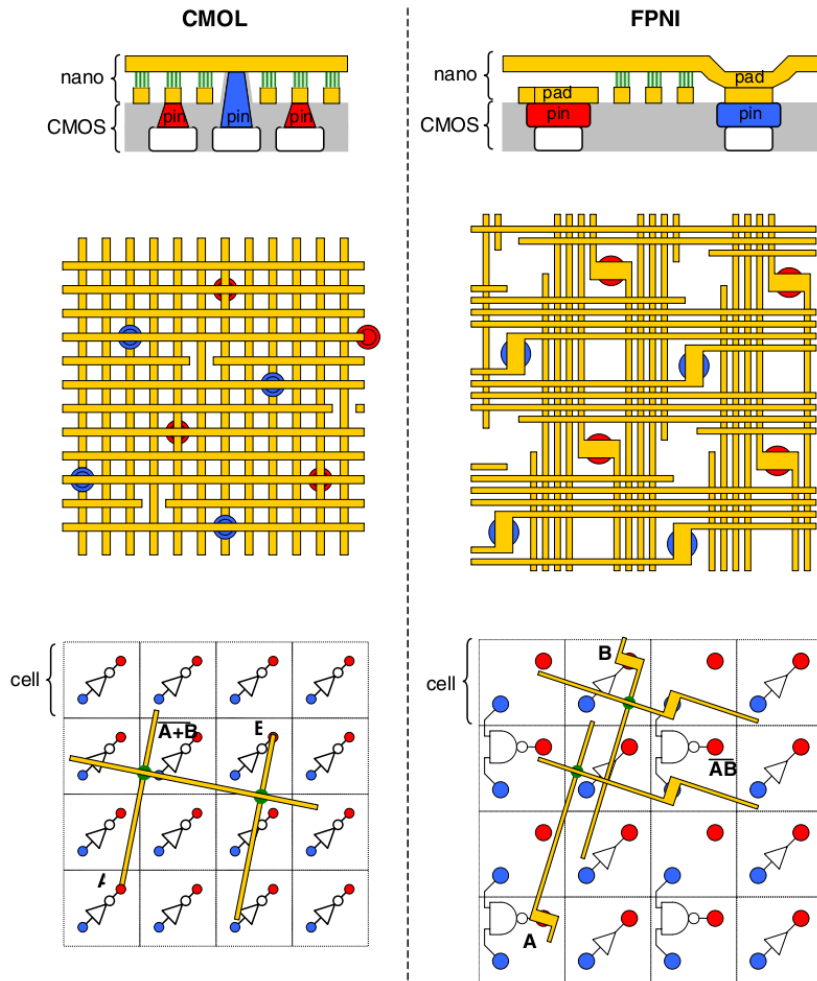


FIGURE 1.6 – Deux approches pour réaliser des circuits hybrides (Snider and Williams, 2007). A gauche, le CMOL consiste à interconnecter avec des nanofils des inverseurs CMOS. En programmant les interconnexions on réalise de la logique OU câblée avec le CMOS fournissant le gain et l'inversion (Strukov and Likharev, 2005). A droite, le FPNI utilise les nanofils uniquement pour réaliser les interconnexions, toute la logique étant implémentée en CMOS (Snider and Williams, 2007).

Nous nous contenterons donc ici de citer quelques publications qui ont eu un impact important. Nous citerons les oscillateurs et circuits chaotiques basés sur le memristor (Muthuswamy and Kokate, 2009) (application suggérée par Chua lorsqu'il a introduit le memristor) et l'électronique analogique programmable (Pershin and Di Ventra, 2010b).

1.3 Traitement des données naturelles

1.3.1 Qu'entend-on par « donnée naturelle » ?

Dans le contexte présent, nous désignerons de manière très générale par donnée naturelle tout type de donnée, principalement sonore et visuelle, pouvant être perçue et interprétée sans effort par un être humain. Il y a notamment la parole, la reconnaissance d'objets, de personnes et de scènes visuelles de la vie courante. Mais cela peut inclure tout signal complexe et difficile à modéliser issu de l'environnement extérieur et perçu par nos sens (du point de vue physiologique), qu'il soit de nature chimique, mécanique, etc...

Concrètement, nous parlons de donnée naturelle par opposition aux données numé-

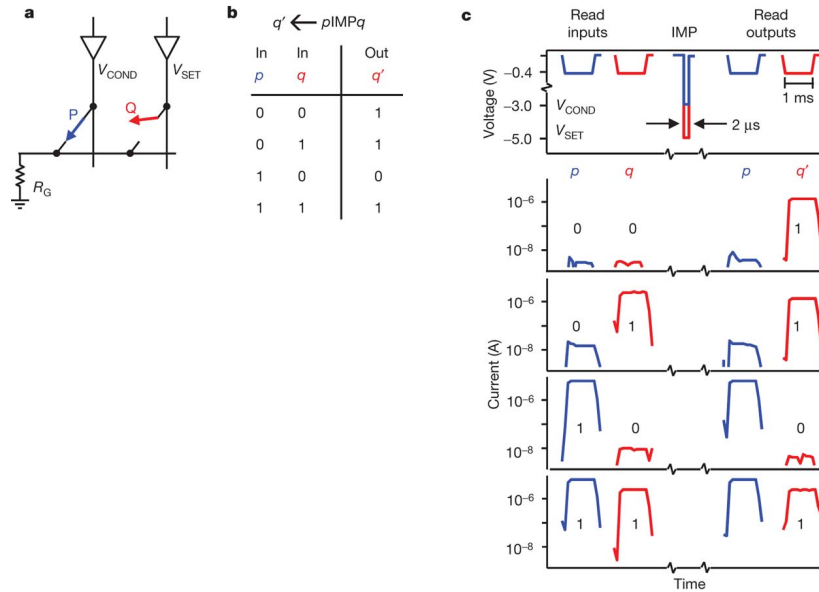


FIGURE 1.7 – Logique d’implication matérielle (Borghetti et al., 2010). (a) L’implication est réalisée en appliquant simultanément les tensions V_{COND} et V_{SET} aux jonctions P et Q, qui entraîne la commutation de Q en fonction de l’état logique de P. (b) La table logique de $q' \leftarrow p \text{ IMP } q$. (c) Tensions appliquées et courants lues aux jonctions P et Q avant et après l’implication.

riques traitées dans un ordinateur, typiquement une série d’instructions. Une différence qualitative existe entre ces deux types de données, qui se mesure au niveau de connaissance requis a priori pour les interpréter. Pour une application de détection de piétons embarquée dans un véhicule (présenté ci-dessous), 10 000 paramètres mémorisés sont nécessaires pour un taux de détection de 97%. De l’ordre de 50 000 paramètres sont nécessaires pour une détection de visages avec un taux de 91.4% (et 10% de faux positifs) sur la base MIT + CMU (Viola and Jones, 2004). Pour la reconnaissance de chiffres manuscrits (base MNIST), plus de 11 millions de paramètres mémorisés sont nécessaires pour atteindre un taux de reconnaissance de 99.7% (Cireşan et al., 2010). Enfin, un taux de reconnaissance de 19.2% sur 10 000 catégories d’objets est atteint avec un milliard de paramètres (soit à raison de 8 bits par paramètre, près de 1 Go de connaissance a priori) (Le et al., 2011). Concernant la reconnaissance de parole, le modèle de Markov caché – «Hidden Markov Model» (HMM) est le plus populaire et nécessite également un grand nombre de paramètres, typiquement de l’ordre de quelques millions pour une langue (Povey et al., 2011), avec des performances encore à l’heure actuelle en retrait par rapport à un être humain. L’amélioration des performances en reconnaissance de parole passe d’ailleurs par des approches incluant des données visuelles sur l’orateur (tel que le mouvement des lèvres) (Potamianos et al., 2003).

Ces chiffres de l’état de l’art montrent que même des algorithmes et des codages optimisés, sur des architectures hiérarchiques, nécessitent une base de connaissance importante pour traiter des données naturelles. En plus de la grande quantité de mémoire, leur traitement nécessite idéalement des accès aux données parallèles et simultanés. Et la tendance pour l’amélioration des performances des systèmes de reconnaissance passe invariablement par une augmentation de la connaissance a priori. Elle est déjà considérable dans l’état de l’art actuel, mais la comparaison en nombre de paramètres avec le nombre de synapses contenues dans le cerveau humain (entre 10^{14} et 10^{15}) reste dérisoire. Le cortex visuel primaire (c’est-à-dire la couche V1 du système visuel) occupe à lui seul un volume proche de 1% du cerveau (Leuba and Kraftsik, 1994), soit proportionnellement un nombre de synapses de 10^{11} à 10^{12} . Même s’il s’agit là d’une

estimation approximative (le cerveau n'est pas homogène), cela reste deux à trois ordres de grandeurs plus élevé en nombre de paramètres comparé à l'état de l'art. Cela suggère fortement que pour approcher les performances de l'humain, cette tendance doit encore largement se poursuivre dans le futur.

1.3.2 Architectures classiques

Par architecture classique, on entend les systèmes implémentés avec de la technologie CMOS, généralement constitués d'un algorithme tournant sur des unités de calcul de type «Central Processing Unit» (CPU) ou «Graphics Processing Unit» (GPU). On s'intéresse plus particulièrement aux systèmes embarqués, pour lesquels la quantité de mémoire et l'accès aux données est une problématique prépondérante. Pour illustrer cette problématique, une architecture de vision de l'état de l'art pour la détection de piéton est prise en référence. Ce type d'architecture a été introduit pour la détection et la reconnaissance de visages (Viola and Jones, 2004). La figure 1.8 présente le système, développé au DIASI (Département Intelligence Ambiante et Systèmes Interactifs) du CEA LIST (Yon et al., 2003). L'étape de détection et classification est de loin la plus exigeante en termes de ressources calculatoires nécessaires (plus de 80% des besoins du système malgré un compromis sur les performances en détection, plus de 95% pour des performances optimales). Le détail de cette étape est expliqué sur la figure 1.9. L'image est analysée avec des boîtes aux échelles et aux positions pouvant correspondre à celles d'un piéton, appelée «Region Of Interest» (ROI). Tout a été fait dans ce système pour réduire au maximum la quantité de données à traiter, en utilisant des signatures calculées en peu d'opérations grâce à une représentation intégrale de l'image (Viola and Jones, 2004). La zone de recherche dans l'image est également restreinte pour l'invariance en translation et en échelle. La classification reste pourtant le goulot d'étranglement car elle nécessite l'accès à énormément de connaissance à priori, qui correspond à la base de signature apprise hors ligne. Cela nécessite une quantité importante d'accès mémoire dont la concurrence est limitée par le bus série entre la mémoire et les unités de calcul. C'est le «bottleneck» de Von Neumann (Backus, 1978).

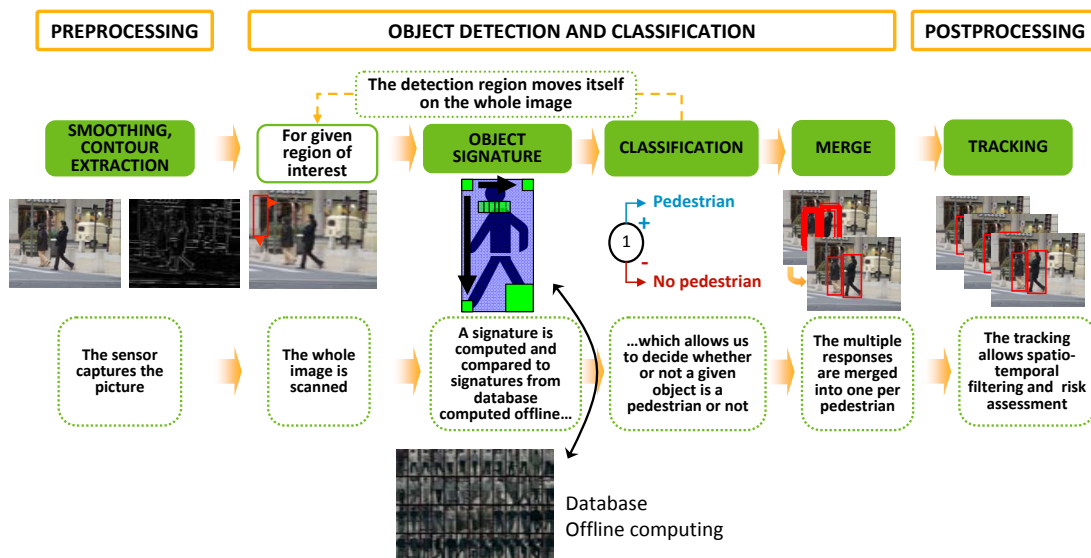


FIGURE 1.8 – Architecture de vision moderne pour la détection de piéton développée au DIASI. L'architecture se décompose en une étape de pré-traitement (extraction de contours et calcul des images intégrales), une étape d'extraction de signatures et de classification et une étape post-traitement pour le suivi de trajectoires.

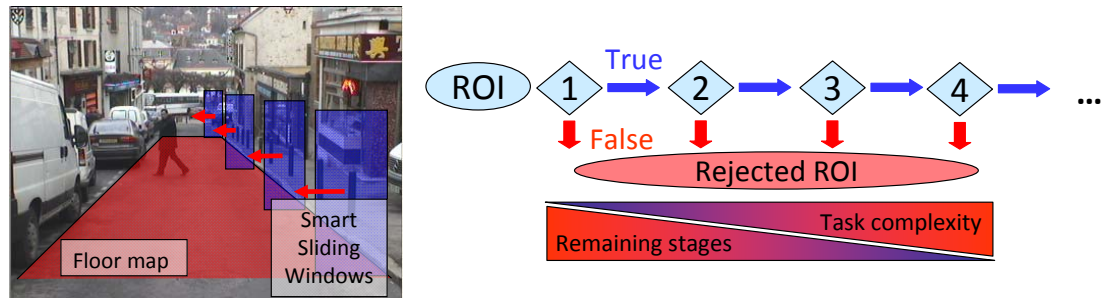


FIGURE 1.9 – Détail de l'étape de détection/classification. L'image est analysée avec des boîtes aux échelles et aux positions pouvant correspondre à celles d'un piéton, appelées ROIs (à gauche). Il y a 1 612 800 ROIs à analyser. Pour chaque ROI, une succession de phases de classification détermine si un piéton est présent ou non. Si à une phase donnée, aucun piéton n'est détecté, la ROI est rejeté, sinon on passe à la phase suivante. La complexité calculatoire des phases successives va en augmentant : pour chaque phase, un certain nombre de signatures sont calculées et comparées avec une base apprise hors ligne. Une signature correspond à un vecteur de l'intensité pour différentes orientations des contours dans une sous-région de la ROI. Pour chaque ROI retenue (présence d'un piéton), environ 1000 signatures auront été testées.

Les systèmes basés sur ou dérivés de réseaux de neurones, implémentés avec des architectures classiques se heurtent de plein fouet à cette problématique, malgré un effort de recherche conséquent dans les années 1990 (Ishii et al., 1992; Kosaka et al., 1995; Tam et al., 1992). Aucune technologie mémoire actuelle ne permet de lire et pondérer/accumuler simultanément des milliers de valeurs efficacement. Cela correspond à l'opération multiplie puis accumule – «Multiply-Accumulate» (MAC) qui est courante dans les processeurs de signal numérique – «Digital Signal Processors» (DSPs) et dont le nombre d'occurrences dans un algorithme de traitement d'image est souvent une bonne mesure de sa complexité. Il est possible d'implémenter des synapses artificielles en technologie CMOS sans passer par un bus mémoire (Hynna and Boahen, 2006; Indiveri et al., 2011). Mais celles-ci nécessitent plusieurs dizaines de transistors et ne permettent donc pas d'atteindre une densité d'intégration et une efficacité énergétique qui les rendent concurrentielles avec de la mémoire traditionnelle. Plus fondamentalement, on retiendra que n'importe quelle structure de classifieur faisant appel à de la connaissance à priori consiste en le calcul de signatures ou l'extraction de motifs et leur comparaison avec des données mémorisées.

1.3.3 Apprentissage versus programmation

La notion d'apprentissage est essentielle dans les systèmes de classification de données naturelles, qu'il s'agisse d'un algorithme supervisé, non supervisé ou plus généralement de la résolution d'un problème d'optimisation. Dans les applications pratiques (telles que la détection de piéton), cet apprentissage se fait hors ligne, sur une base d'exemples, puis le résultat est chargé en mémoire ou programmé dans le système final. Le processus d'apprentissage est en effet l'une des pires démonstration du «bottleneck» de Von Neuman. Si l'on prend pour référence la règle d'apprentissage classique de la rétro-propagation du gradient dans les réseaux de neurones (Rumelhart et al., 1986), celle-ci consiste à venir charger un poids, calculer sa mise à jour, puis la sauvegarder en mémoire et cela pour toutes les synapses du réseau pour chaque présentation d'un exemple d'apprentissage. Ainsi, le temps d'apprentissage pour les systèmes actuels de reconnaissance atteint aisément plusieurs jours, voire plusieurs semaines sur une machine de bureau standard lorsque le nombre de paramètres et la base d'exemple fait plusieurs millions d'éléments.

Un apprentissage hors ligne puis une programmation de la mémoire du système de traitement suppose toutefois que celle-ci ne contienne aucun défaut et peu de variabilité. Ces contraintes induisent un coût significatif pour l'intégration des technologies mémoires émergentes à base de nano-dispositifs memristifs. L'augmentation de la densité d'intégration et des performances, qui se fait au détriment de la reproductibilité des caractéristiques des dispositifs individuels, pourrait être rapidement stoppée par le coût et la complexité des circuits à mettre en œuvre pour gérer ces défauts dans le monde numérique (Snider, 2007). Une solution pourrait être de caractériser au préalable le circuit, puis de réaliser un apprentissage tenant compte de ses caractéristiques et défauts pour le programmer ensuite. Mais cette approche ne semble pas réaliste pour une fabrication en masse, compte tenu du coût de l'apprentissage.

Une architecture intrinsèquement parallèle à base de dispositifs memristifs pourrait permettre de surmonter ces difficultés en se programmant par apprentissage. Plutôt que de réaliser l'apprentissage hors ligne, celui-ci se ferait directement sur les dispositifs memristifs, et tiendrait donc compte des défauts et de la variabilité. L'apprentissage pourrait constituer de fait la méthode la plus efficace pour programmer de manière robuste et précise des dispositifs aux caractéristiques fortement dispersées et variables d'un changement d'état à l'autre (Alibart et al., 2012b). En outre, l'apprentissage permettrait d'exploiter efficacement la mémoire analogique des dispositifs memristifs. Il pourrait remplacer les cycles de programmation-vérification, seule solution pour de la programmation multi-niveaux robuste, même pour des dispositifs au stade industriel tels que la PCM et la CBRAM (Papandreou et al., 2011; Schrogmeier et al., 2007).

Enfin, l'apprentissage en ligne ouvre potentiellement la voie à la fabrication de systèmes de traitement génériques et adaptables au problème à résoudre in situ. De tels systèmes pourraient implémenter des algorithmes d'apprentissage non supervisé, pour l'extraction automatique des motifs et des corrélations optimales (Olshausen and Field, 1996). Les travaux qui seront présentés par la suite explorent en profondeur cette perspective.

1.4 Ce que nous apprennent les neurosciences

De nombreux systèmes pour la perception s'inspirent des réseaux de neurones biologiques, en particulier du système visuel humain pour la reconnaissance d'éléments visuels. L'organisation hiérarchique de celui-ci se retrouve dans de nombreux algorithmes : filtrage, extraction de contours et de points d'intérêt, classification... Malgré tout, la manière d'implémenter ces fonctionnalités en dehors des neurosciences est généralement très éloignée de la biologie. Les premiers réseaux de neurones formels, de type perceptron (Rosenblatt, 1958), en sont un exemple. Il s'agit de réseaux purement statiques, qui ne font pas intervenir le temps dans le codage de l'information, contrairement aux réseaux impulsifs biologiques. Ce sont les limitations de ces modèles, ainsi qu'une meilleure compréhension du codage de l'information et de l'apprentissage dans le cerveau qui poussent aujourd'hui à se rapprocher des neurosciences (Maass, 1997), comme nous allons le voir dans cette partie. Le mode de programmation impulsif des dispositifs memristifs dans un «crossbar» est une autre motivation majeure qui pousse à s'orienter vers un codage temporel de l'information (Zamarreño-Ramos et al., 2011).

1.4.1 Codage de l'information

L'information transite dans le cerveau sous la forme d'impulsions électriques, ou potentiels d'action. Le mécanisme de transduction des stimuli visuels et sonores en

trains d'impulsions est donc un premier élément essentiel pour la compréhension du codage et du traitement de l'information. La figure 1.10 explique de manière brève et simplifiée le principe de fonctionnement de la rétine. Les cellules sensibles à la lumière (les cônes et les bâtonnets) propagent un signal vers les cellules bipolaires, sensibles essentiellement aux changements de luminosité, signal qui est converti en un train d'impulsions dans les cellules ganglionnaires de la rétine. La figure 1.11 montre plus en détail l'activité typique des cellules ganglionnaires pour différentes stimulations de leur champ récepteur. Contrairement à une caméra vidéo, la rétine ne capture donc pas une succession d'images statiques, mais répond localement et de manière asynchrone aux variations d'intensité lumineuse. Un changement complet de plan visuel (par exemple lors de la lecture d'un album photo) induit toutefois logiquement une réponse globale synchrone sous la forme d'une vague de trains d'impulsions, sur une durée de l'ordre de 20 à 40 ms. En outre, la rétine compresse l'information d'un facteur 100, puisque le nombre de cellules photosensibles est d'environ 100 millions alors que le nombre de fibres nerveuses constituant le nerf optique n'est que d'un million.

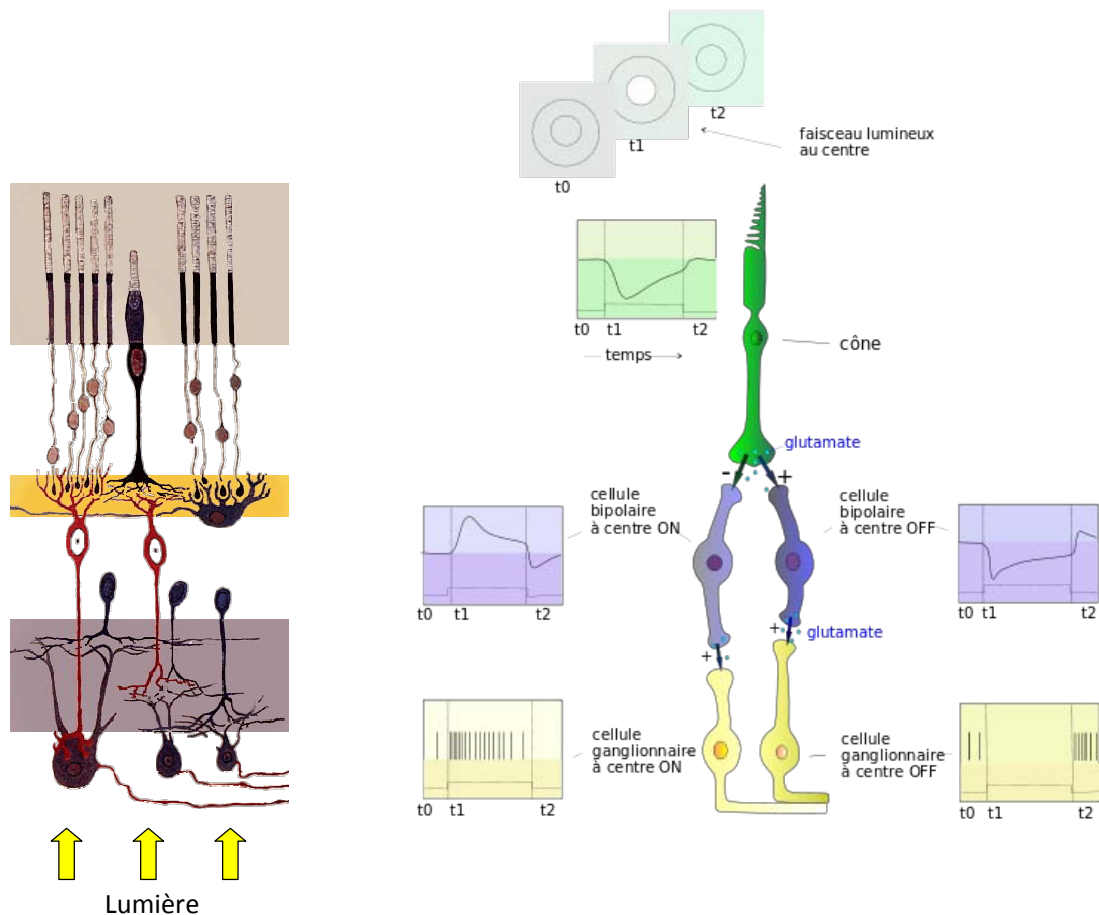


FIGURE 1.10 – Circuits impliqués dans la production des réponses des champs récepteurs de la rétine (Purves et al., 2005). Le fond de l'œil est en haut, la lumière venant d'en bas (topologie similaire à un capteur CMOS classique, où la lumière doit traverser les couches d'interconnexions en métal pour atteindre le capteur). Sur la figure de gauche, les cônes et les bâtonnets sont en haut et le signal se propage de haut en bas jusqu'aux fibres nerveuses du nerf optique, en passant par les cellules bipolaires (couche jaune) et les cellules ganglionnaires. La figure de droite est une représentation schématique de celle de gauche, qui montre l'évolution des potentiels d'action des différentes cellules pour un changement de luminosité.

Malgré une très bonne compréhension du fonctionnement de la rétine, le codage de l'information transitant dans le nerf optique continue de faire débat (Van Rullen

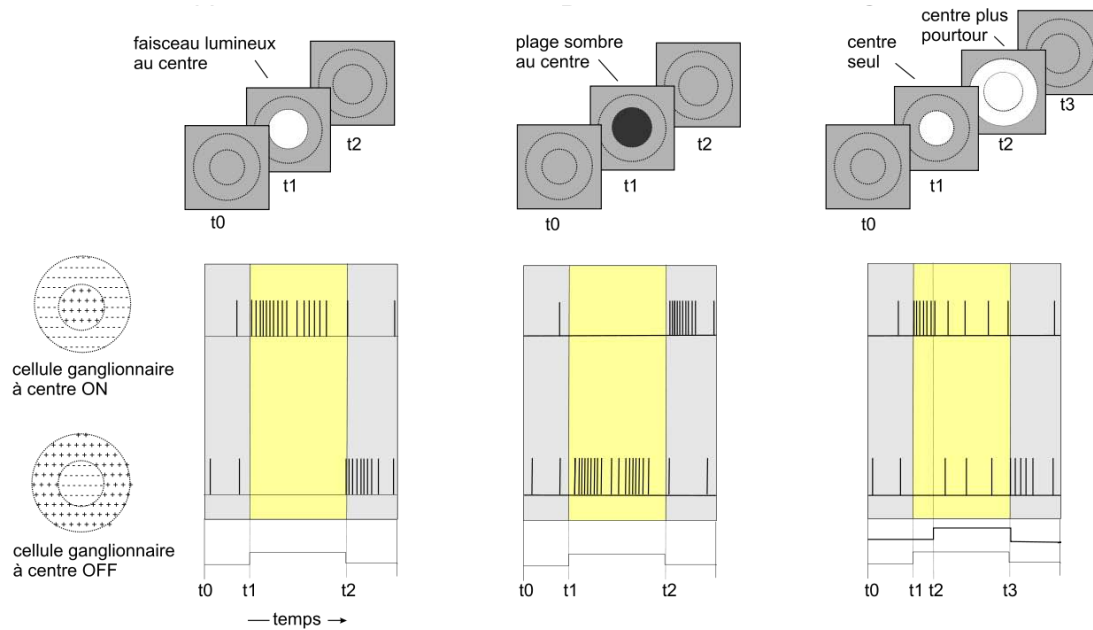


FIGURE 1.11 – Réponse des cellules ganglionnaires ON et OFF pour différentes stimulations de leur champ récepteur (Purves et al., 2005). L'activité impulsionnelle en sortie des cellules dépend de leur éclaircissement. Celles-ci sont particulièrement sensibles aux changements de luminosité.

and Thorpe, 2001). La manière dont cette information est exploitée dans les différentes couches du système visuel est en effet encore mal comprise. Le modèle le plus courant pour décrire les trains d'impulsions observés est un modèle probabiliste de Poisson, caractérisé par une fréquence moyenne. Toutefois, la capacité d'un être humain de reconnaître un objet ou une personne en à peine plus de 200 ms suggère que l'information ne peut pas être codée simplement fréquemment. Les fréquences moyennes des potentiels d'action dans le cerveau sont en effet de l'ordre de 1 à 100 Hz (Thorpe et al., 1996). Simon Thorpe notamment défend l'idée d'un codage basé sur l'ordre d'arrivée des potentiels d'action et propose un modèle basé sur ce principe pour la reconnaissance d'objets (Thorpe, 1990). Celui-ci est mis en œuvre avec succès par SpikeNet Technology pour leur solution de vision logicielle (Delorme et al., 1999; Thorpe et al., 2004).

1.4.2 Mécanismes d'apprentissage

Les capacités d'apprentissage et de généralisation du cerveau sont formidables et restent hors de portée des connaissances et de la technologie actuelle. Néanmoins, les mécanismes de plasticité des synapses sont de mieux en mieux connus, avec les premières observations dans les années 1990 de la modification de l'efficacité synaptique en fonction de la coïncidence entre les impulsions pré- et post-synaptiques (Markram et al., 1997; Bi and Poo, 1998). Cette plasticité synaptique induite par les potentiels d'action est appelée «Spike-Timing-Dependent Plasticity» (STDP). Ce mécanisme est aujourd'hui considéré comme fondamental pour expliquer les capacités d'apprentissage et d'adaptation du cerveau (Dan and ming Poo, 2004). Et ce, même s'il ne s'agit que d'un modèle possible pour interpréter les mesures biologiques, qu'il serait bien difficile d'établir formellement. En particulier, il offre une base solide pour comprendre les phénomènes de potentialisation à long terme – «Long-Term Potentiation» (LTP) et de dépression à long terme – «Long-Term Depression» (LTD). STDP est largement utilisée, bien qu'avec de nombreuses variations, à la fois dans les neurosciences computationnelles (Bohte and Mozer, 2007; Izhikevich and Desai, 2003) et dans l'apprentissage automatique (Nessler et al., 2010; Masquelier and Thorpe, 2007; Gupta and

Long, 2007). STDP est un raffinement de la règle de Hebb avec une prise en compte des aspects temporels. La figure 1.12 présente des mesures expérimentales sur des synapses biologiques mettant en évidence différentes formes de STDP (asymétrique à gauche et symétrique à droite).

La manière exacte dont STDP est utilisée pour apprendre dans le cerveau reste néanmoins encore floue, même si cette règle d'apprentissage arrive à concurrencer des algorithmes classiques tels que celui de la rétro-propagation du gradient, lorsque le codage des données est adapté. La figure 1.13 présente une démonstration des capacités de STDP à extraire un motif temporel noyé dans du bruit. Les synapses activées juste avant le déclenchement du neurone sont renforcées, tandis que celles activées trop tardivement sont affaiblies. Le neurone renforce ainsi les synapses dont l'instant d'activation est corrélé avec son déclenchement. Dans le cas du motif répétitif dans du bruit, dès lors que le neurone s'active durant le motif, celui-ci va renforcer les synapses activées par le motif et aura plus de chance de se réactiver lors de la prochaine présentation de celui-ci. Dans le même temps, puisque les synapses activées plus tardivement sont affaiblies, le neurone aura moins de chance de se déclencher sur une portion de bruit. Après une période d'apprentissage, le neurone parvient à devenir exclusivement sélectif au motif.

En résumé, la conséquence de la règle d'apprentissage STDP est de renforcer le poids des synapses qui contribuent de manière répétée à activer le neurone, le rendant ainsi sensible à un motif répétitif dans les stimuli d'entrée. Par motif, on entend ici un paquet d'impulsions (ou potentiels d'action) fortement corrélées temporellement.

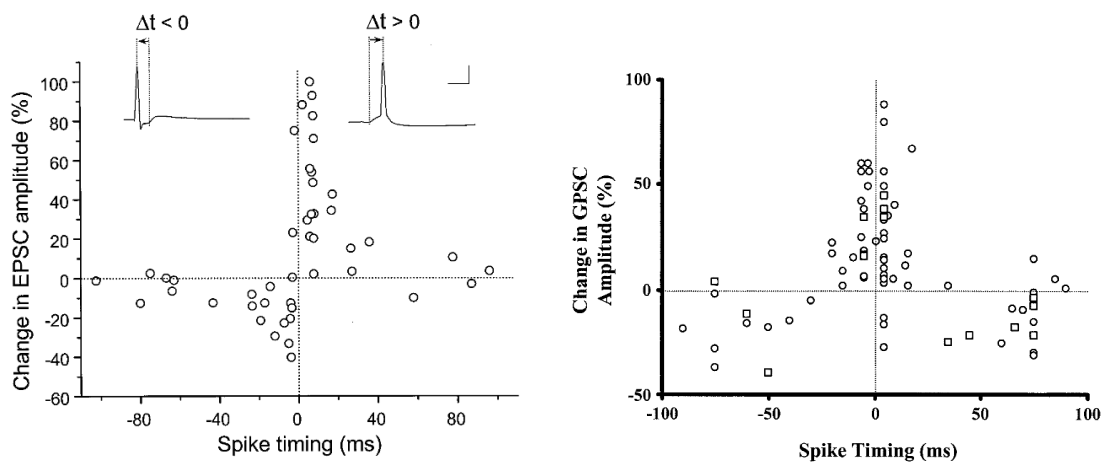


FIGURE 1.12 – STDP observée en biologie. A gauche, le pourcentage de modification du courant post-synaptique excitateur – «Excitatory Postsynaptic Current» (EPSC) 20 à 30 minutes après un train de 60 impulsions à 1 Hz pré-post corrélées (Bi and Poo, 1998). A droite, une mesure similaire sur la modification du courant post-synaptique dans des synapses GABAergiques (10 à 20 minutes après un train d'impulsions de 30 s à 5 Hz) (Woodin et al., 2003).

1.4.3 Architectures biologiques

L'un des systèmes corticaux les plus étudiés et les mieux connus est le cortex visuel, notamment chez les primates. Il se décompose en plusieurs couches, les trois premières, V1, V2 et V3 faisant largement consensus dans les neurosciences. Il s'agit d'un réseau bien hiérarchisé et dans lequel l'information se propage principalement dans un sens («feedforward») (Thorpe and Imbert, 1989; Thorpe et al., 1996). Fonctionnellement, la première partie de la couche V1 consiste en des filtres spatio-temporels sélectifs et bien localisés. Ces champs récepteurs sont typiquement modélisés par des filtres de Gabor (Daugman, 1985), sensibles à l'orientation spatiale et temporelle (mouvement). Il est

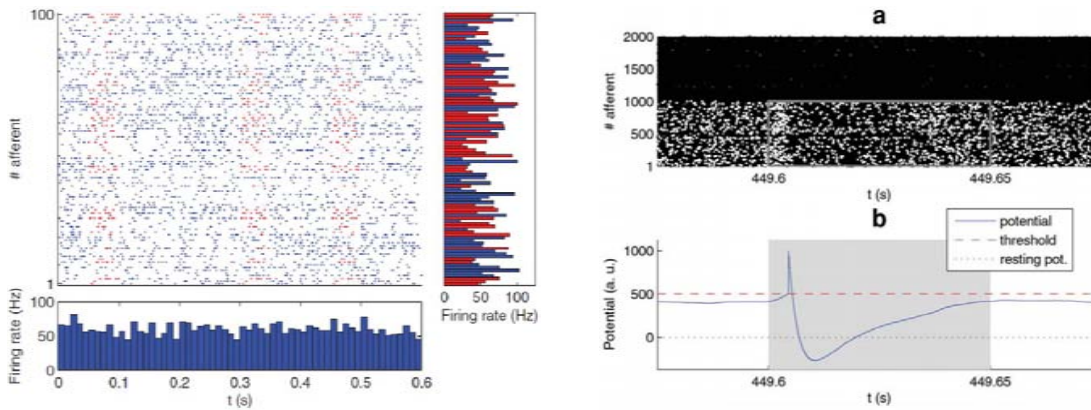


FIGURE 1.13 – Extraction non supervisée d'un motif répétitif dans du bruit (Masquelier et al., 2008). A gauche, le motif spatio-temporel d'entrée, avec en rouge un motif répétitif de 50 ms sur 500 des 1000 entrées (seules 100 entrées sont représentées). Les fréquences moyennes sur les entrées (encart à droite) ou sur des intervalles de temps de 10 ms (encart en bas) ne permettent pas de caractériser le motif répétitif, ce qui exclut une détection purement fréquentielle. A droite, l'état des synapses et du neurone de sortie après apprentissage. Les synapses renforcées sont celles activées au début du motif répétitif (points blancs dans l'encart en haut), qui induisent une brusque augmentation du potentiel de membrane (à partir de 449.6 s dans l'encart en bas) suffisante pour déclencher le neurone et ainsi détecter le motif répétitif.

remarquable que ce type de champ récepteur peut émerger naturellement avec STDP sur une base de stimuli visuels suffisamment large (Delorme et al., 2001). La couche V2 peut encoder des formes plus complexes telles que des combinaisons d'orientations, prémisses de la détection de contour et de la segmentation de surface (Anzai et al., 2007).

Cette organisation en couches n'est pas incompatible avec un codage basé sur l'ordre d'arrivée des impulsions, qui pourraient se propager par vagues successives d'une couche à l'autre (VanRullen and Thorpe, 2002). Chaque couche peut manipuler l'information de rang (les instants d'émission des impulsions) en fonction des sélectivités des neurones ou de l'inhibition latérale par exemple.

La figure 1.14 présente un exemple d'architecture bio-inspirée s'inspirant des couches V1 et V2 du système visuel humain, pour la détection de visages. Les couches S1 et C1 sont sensibles aux contours, tandis que des formes complexes sont apprises sur la couche S2 de manière non supervisée avec STDP. Certaines caractéristiques visuelles apprises par la couche S2 sont illustrées sur la figure 1.15. Les résultats de cette étude montrent que les caractéristiques extraites avec STDP sont analogues à celles qu'il serait possible d'obtenir avec la règle de Hebb et un codage non temporel. L'apprentissage dans ce système est compétitif : un neurone qui se déclenche inhibe ses voisins. Ce mécanisme de type «Winner-Take-All» (WTA) est présent également dans les systèmes biologiques et permet à chaque neurone de se spécialiser dans un motif différent.

De la formation de champs récepteurs complexes dans les premières couches du système visuel à la reconnaissance de personnes et d'objets et leur association avec d'autres concepts plus abstraits, il reste un large champ d'investigation pour les neurosciences. A ce niveau, une découverte importante a vu le jour récemment, celui des «concept cells» (Quian Quiroga et al., 2005). Des neurones ultra sélectifs à des objets et des personnes ont été observés chez l'être humain. Par exemple, un neurone s'activant à la vue de Jennifer Aniston mais pas d'autres femmes semblables, ou encore un neurone s'activant à la fois à la vue d'une personne et à l'évocation de son nom (Roy, 2011). Ces cellules pourraient être une clé pour comprendre les processus de traitement en amont et pourquoi pas certains mécanismes de la pensée en aval. Elles ramènent en tout cas

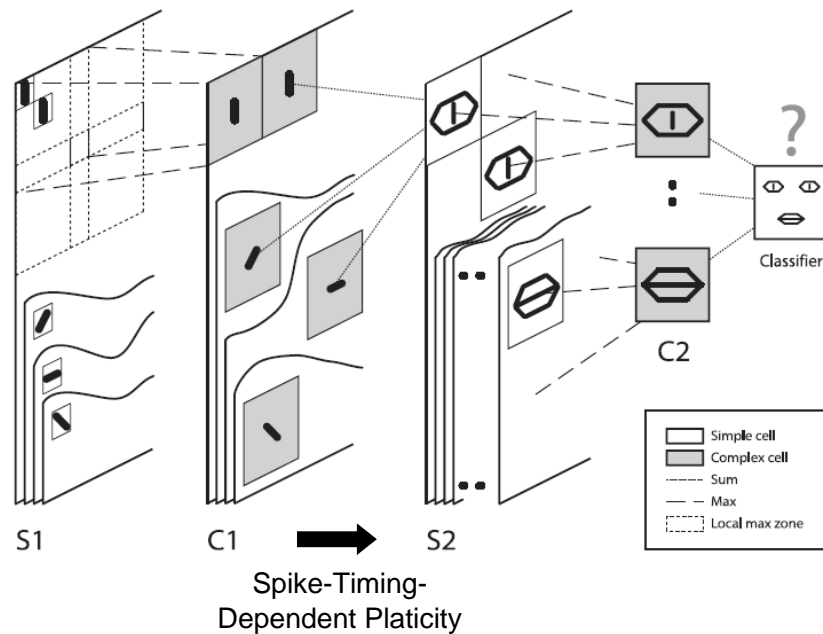


FIGURE 1.14 – Principe d'extraction non supervisée de caractéristiques visuelles avec STDP (Masquelier and Thorpe, 2007). Il s'agit d'un réseau de neurones «feedforward» constitué de 5 couches. S1 détecte les contours au moyen de convolutions, C1 sous-échantillonne S1 à l'aide de l'opérateur MAX, tandis que S2 est sensible à certains motifs de contours. L'extraction de motifs de la couche C1 à S2 se fait grâce à l'algorithme d'apprentissage STDP, qui identifie de manière non supervisée les caractéristiques pertinentes des images sur lesquels se fait l'apprentissage (par exemple un œil ou une bouche dans le cas de détection de visages comme cela est suggéré par la figure). Enfin, C2 prend la réponse maximale de S2 pour effectuer une classification (présence ou non de visage).



FIGURE 1.15 – Reconstruction des caractéristiques visuelles extraites avec STDP, pour l'apprentissage de visages et de motos (Masquelier and Thorpe, 2007). Certaines cellules ont développé une sélectivité aux visages et d'autres aux motos.

les capacités de traitement des données naturelles du cerveau dans une dimension très concrète, matérielle et mesurable !

1.5 Discussion et perspectives

Dans ce chapitre, nous avons introduit les dispositifs memristifs de manière générale, sans se focaliser sur une technologie particulière, puisque cela est l'objet du chapitre

suivant. Nous avons notamment décrit les caractéristiques principales de ces dispositifs et les problèmes que cela pose pour leur intégration dans des architectures de calcul numériques classiques. A l'opposé, nous avons vu en quoi ces dispositifs avaient un fort potentiel pour le traitement de données naturelles, pour lesquelles une répartition spatiale des calculs est plus adaptée qu'une répartition temporelle, propre aux architectures des ordinateurs actuels. Les problématiques du «bottleneck» de Von Neumann et de la quantité de connaissance à priori nécessaire à un système de reconnaissance ont été évoquées. La programmation de nano-composants par nature peu fiables et très variables pour de telles architectures pourrait être résolue par un mécanisme d'apprentissage. Enfin, un état de l'art bref et choisi sur ce que nous apprennent les neurosciences dans le fonctionnement des réseaux de neurones biologiques met en exergue l'adéquation potentielle entre les dispositifs memristifs et les architectures bio-inspirées pour le traitement de données naturelles.

Nous espérons avoir posé dans ce chapitre les bases essentielles pour la compréhension de ces travaux et pour en saisir leur pertinence pour le futur des architectures de traitement de données. Les idées présentées ici suggèrent de s'orienter vers des systèmes véritablement en rupture avec l'existant, avec des gains potentiels de plusieurs ordres de grandeurs sur la consommation énergétique, la surface silicium et leurs performances fonctionnelles. Les résultats qui vont suivre visent à confirmer la viabilité de ces approches et évaluer dans la mesure du possible les gains réels qu'elles peuvent nous apporter. Le potentiel est énorme, mais la réalité du monde industriel et de l'évolution des systèmes de traitement vis-à-vis des réseaux de neurones artificiels nous invite à rester prudent !

Chapitre 2

Technologies mémoires émergentes et memristivité

Sommaire

2.1	Introduction	27
2.2	Nanoparticle-Organic Memory transistor (NOMFET)	28
2.2.1	Description	28
2.2.2	Modèle fonctionnel	30
2.2.3	Extraction des paramètres du modèle	31
2.2.4	Implémentation et simulations	34
2.2.5	Dépression synaptique et contrôle du gain	36
2.3	Mémoire à changement de phase (PCM, PRAM ou PCRAM)	38
2.3.1	Description	38
2.3.2	Caractérisation expérimentale	40
2.3.3	Modélisation	40
2.4	Conductive-Bridging RAM (CBRAM)	46
2.4.1	Description	47
2.4.2	Caractérisation expérimentale	48
2.5	Resistive RAM (RRAM ou ReRAM)	49
2.5.1	Description	49
2.5.2	Exemples de caractéristiques	51
2.6	Dispositifs à jonction tunnel	52
2.7	Discussion et perspectives	53

2.1 Introduction

Ce chapitre présente en détail les trois technologies auxquelles nous avons eu accès dans le cadre de collaborations et sur lesquelles nous avons basé nos travaux : le «Nanoparticle-Organic Memory Field Effect Transistor» (NOMFET), la mémoire à changement de phase – «Phase-Change Memory» (PCM) et la «Conductive-Bridging RAM» (CBRAM). Le NOMFET est fabriqué par l’Institut d’Électronique, de Microélectronique et de Nanotechnologie (IEMN) à Lille, dans l’équipe de Dominique Vuillaume. La PCM est fabriquée par le Laboratoire d’électronique des technologies de l’information du Commissariat à l’énergie atomique et aux énergies alternatives (CEA) (CEA LETI) à Grenoble en collaboration avec STMicroelectronics et la CBRAM est fabriquée par Altis Semiconductor en collaboration avec le CEA LETI. Ces deux dernières technologies sont par ailleurs considérées comme les plus matures pour une intégration à l’échelle industrielle.

Dans le cadre des technologies memristives, il n'est pas possible de faire l'impasse sur la technologie de mémoire résistive – «Resistive RAM» (RRAM), qui constitue également un champ d'exploration prometteur, même si celle-ci n'a pas encore atteint le niveau de maturité des deux technologies précédentes. Une partie présentant les technologies de type RRAM est donc également présente. Les dispositifs à jonction tunnel, dont la mémoire magnétique – «Magnetic RAM» (MRAM), qui n'ont pas fait l'objet d'étude dans ces travaux, sont également brièvement évoqués.

L'objectif de ce chapitre est d'introduire pour chaque technologie les caractérisations expérimentales utiles pour une utilisation synaptique de celles-ci, ainsi que de donner un aperçu des mécanismes physiques sous-jacents. Un modèle fonctionnel ou semi-physique est également proposé pour le NOMFET et la PCM. Des architectures pour le traitement de données naturelles capables de tirer partie de ces technologies seront introduites dans le chapitre «Étude d'architectures neuro-inspirées nouvelle génération pour le traitement de données naturelles» et des implémentations utilisant les caractérisations et les modèles introduits ci-après seront proposées dans le chapitre «Implémentations matérielles sur des technologies memristives».

2.2 Nanoparticle-Organic Memory transistor (NOMFET)

Le NOMFET est un dispositif memristif particulier, puisqu'il est volatile et n'est donc utilisable qu'en régime dynamique. Il possède néanmoins des caractéristiques dynamiques remarquablement similaires à celles d'une synapse biologique (Novembre et al., 2008; Alibart et al., 2010). Montrer qu'il est possible d'implémenter des fonctions neuro-inspirées telles que la «liquid state machine» (Maass et al., 2002) ou de l'apprentissage «Spike-Timing-Dependent Plasticity» (STDP) (Markram et al., 1997) sur un tel dispositif constitue un travail original et en rupture par rapport aux propositions architecturales sur les autres dispositifs memristifs. Une technique spécifique a été développée pour la modélisation du comportement dynamique du NOMFET.

Dans cette partie, nous introduisons le NOMFET et son comportement synaptique et proposons un modèle fonctionnel que se base en partie sur les mécanismes physiques du dispositif. Celui-ci est plus réaliste qu'un modèle purement logique et plus rapide à simuler qu'un modèle compact (Meixner et al., 2008) pour de l'exploration architecturale. Une méthodologie pour extraire les paramètres du modèle à partir des données expérimentales est détaillée. Une implémentation du modèle en Verilog-A (Agi, 2007) sur le simulateur Spectre de Cadence (Cad, 2008a) est présentée avec des résultats de simulation. Enfin, nous simulons avec ce modèle une fonction de contrôle du gain synaptique.

Ces travaux ont été publiés en collaboration avec l'IEMN dans (Bichler et al., 2010) (Bichler et al., 2010).

2.2.1 Description

Le NOMFET est constitué de trois terminaux, comme le transistor «Metal Oxide Semiconductor» (MOS) conventionnel (voir figure 2.1) : drain (D), source (S) et grille (G). L'électrode de grille est fabriquée avec une configuration grille par le dessous, couverte de 200 nm de SiO₂. Les terminaux D et S sont en or avec un intervalle inter-électrode de 0.2 à 20 μm . Des nano-particules (NPs) d'or (diamètre de 5, 10 ou 20 nm) sont immobilisées chimiquement sur la surface de l'intervalle inter-électrode avant la déposition du pentacène. Le pentacène est un semi-conducteur organique de type p (Novembre et al., 2008; Alibart et al., 2010). La conduction électrique entre les terminaux D et S est assurée par les trous, qui sont créés à l'interface avec l'oxyde de

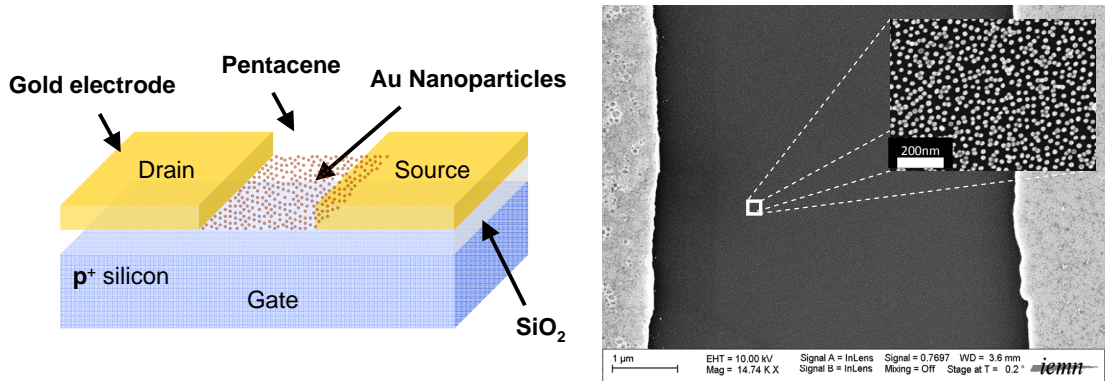


FIGURE 2.1 – Structure physique du transistor NOMFET. Celui-ci est constitué d’une électrode de grille dopée p+ en dessous, couverte d’oxyde de silicium. Les électrodes de source et de drain sont en or et les NPs d’or sont déposées dans l’interval inter-électrode avant la déposition de pentacène.

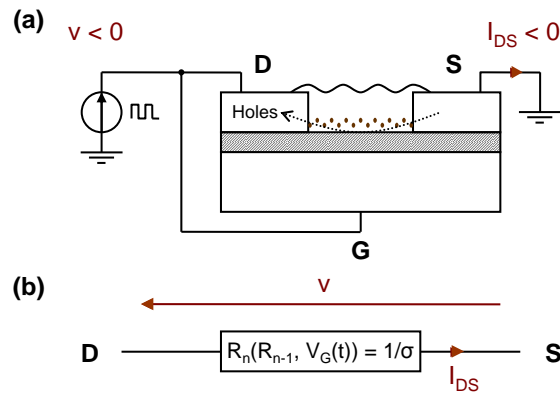


FIGURE 2.2 – (a) Configuration du dispositif utilisé pour établir notre modèle, (b) circuit électrique équivalent (en régime linéaire). La tension d’entrée v est une séquence d’impulsions d’amplitude V_{AP} . R_n est la résistance équivalente drain-source à la n -ième impulsion. R_n dépend de la polarisation du dispositif et du niveau de charge des NPs, contenu dans le terme R_{n-1} .

silicium lors de l’application d’une tension de grille négative. Le courant drain-source I_{DS} est négatif. En plus du comportement classique d’un transistor, une tension de grille négative charge positivement les NPs d’or. Cela réduit la conductivité du canal du transistor, car les NPs chargées induisent une interaction électrostatique répulsive entre les trous piégés dans les NPs et ceux accumulés dans le canal de pentacène. Le temps de rétention de la charge dans les NPs peut atteindre plusieurs dizaines de minutes, ce qui confère au NOMFET sa propriété de mémoire à court terme, ainsi que sa memristivité (Novembre et al., 2008).

Le modèle présenté dans la suite a été établi pour une configuration du dispositif à deux terminaux comme illustré sur la figure 2.2. Les électrodes de grille et de drain sont commandées par la même tension d’entrée, une impulsion d’amplitude constante V_{AP} et une fréquence variable (voir figure 2.3). Il y a une compétition entre la charge des NPs ajoutée par les impulsions de grille et la relaxation naturelle de celle-ci. Cela peut induire respectivement une diminution et une augmentation du courant du canal I_{DS} . Lorsqu’une nouvelle séquence d’impulsions d’entrée survient, le NOMFET présente donc un comportement soit dépressif, soit facilitateur, selon l’intervalle entre les impulsions et le niveau de charge des NPs. Le courant I_{DS} est donc dépendant de l’historique du signal d’entrée.

Cette propriété est équivalente au comportement d’une synapse biologique dépres-

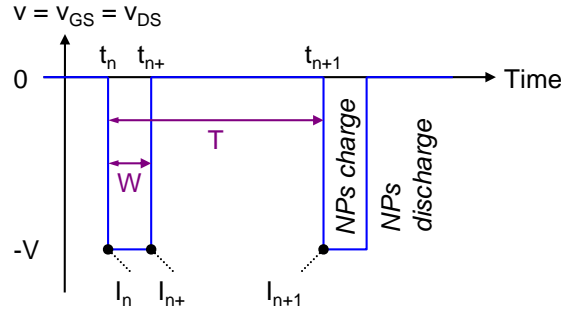


FIGURE 2.3 – Train d’impulsions en entrée sur les électrodes de drain et de grille. I_n est la valeur du courant drain-source I_{DS} à la n -ième impulsion, qui a lieu au temps t_n . La nouvelle valeur I_{n+1} est calculée à partir de I_n , W et T . W est la largeur de l’impulsion précédente et T est l’intervalle entre les impulsions, qui peut être variable.

sive, avec une plasticité à court terme (Abbott et al., 1997). L’amplitude des impulsions transmises représente la quantité de neurotransmetteurs disponible à un temps donné. Elle dépend à la fois de l’historique et de l’activité actuelle du stimulus sur la synapse. Ainsi, l’amplitude des impulsions transmises peut être diminuée ou augmentée en modulant la fréquence de la stimulation. La variation maximale de l’amplitude des impulsions transmises dans nos dispositifs est proche de 30% pour les NOMFETs avec un intervalle inter-électrode de l’ordre du micron. Une variation allant jusqu’à 40% a pu être montrée avec des dispositifs avec un intervalle inter-électrode réduit à 200 nm et une tension de commande de -3 V (Alibart et al., 2010) (voir également la figure 2.6). Des variations de courant plus importantes sont possibles dans d’autres configurations du dispositif.

2.2.2 Modèle fonctionnel

Le modèle intègre à la fois l’effet des charges piégées dans les NPs sur I_{DS} (c’est-à-dire sur la conductivité) et les dynamiques de charge/décharge des NPs. Le courant dans le dispositif est déterminé par l’expression proposée dans (Alibart et al., 2010) :

$$I_{DS} = \sigma \cdot V_{DS} \text{ avec } \sigma = A_0 \cdot e^{\beta \cdot \epsilon_f} \cdot e^{-\beta \cdot \Delta} \quad (2.1)$$

avec $\beta = 1/(k_B \cdot \theta)$, θ est la température, A_0 est un paramètre dépendant de la température, ϵ_f est l’énergie de Fermi fixée par la tension de grille et Δ est la variation de l’énergie de Fermi induite par les charges piégées dans les NPs. Pour une polarisation donnée, une forme simplifiée de l’équation précédente est :

$$I_{DS} = \tilde{I} \cdot e^{-\beta \cdot \Delta} \quad (2.2)$$

où \tilde{I} est un pseudo-courant représentant la conductivité entre les terminaux de drain et de source lorsque les NPs sont à l’équilibre et que les trois terminaux sont reliés à la masse. Par soucis de simplification, on suppose dans notre modèle que dans ce cas les NPs sont complètement déchargées. Cette expression est cohérente avec l’interprétation physique présente dans la partie précédente.

Construction d’un modèle itératif

Le signal appliqué à l’entrée du dispositif est illustré figure 2.3. I_n est le courant drain-source à la n -ième impulsion. De t_n à $t_{n+1} = t_n + W$ (largeur de l’impulsion),

la tension négative V_{GS} charge les NPs. La charge suit une certaine dynamique caractérisée par une fonction empirique $f_C(t)$, prenant ses valeurs entre 0 et 1. Pour une largeur d'impulsion infinie, les NPs tendent à être complètement chargées, induisant une variation de l'énergie de Fermi maximale Δ_{\max} . L'expression du courant à $t = t_{n+}$ est

$$I_{n+} = I_n \cdot f_C(W) + \tilde{I} \cdot e^{-\beta \cdot \Delta_{\max}} \cdot (1 - f_C(W)) \quad (2.3)$$

De t_{n+} à $t_{n+1} = t_n + T$, la charge piégée dans les NPs est relaxée et le courant tend vers \tilde{I} . Comme pour la charge, la dynamique de relaxation est décrite par une autre fonction empirique $f_D(t)$

$$I_{n+1} = I_{n+} \cdot f_D(T - W) + \tilde{I} \cdot (1 - f_D(T - W)) \quad (2.4)$$

L'expression générale de l'évolution du courant I_{DS} entre deux impulsions peut être obtenue en intégrant 2.3 dans 2.4 :

$$I_{n+1} = I_n \cdot f_C(W) \cdot f_D(T - W) + \tilde{I} \cdot \left[e^{-\beta \cdot \Delta_{\max}} \cdot (1 - f_C(W)) \cdot f_D(T - W) + 1 - f_D(T - W) \right] \quad (2.5)$$

Caractéristiques de charge/décharge des NPs

En introduisant deux constantes de temps τ_C et τ_D dans le modèle, une simple fonction exponentielle peut être utilisée pour les fonctions $f_C(t)$ et $f_D(t)$, ce qui est suffisant pour simuler les dynamiques de charge et de décharge avec une bonne précision :

$$f_C(t) = e^{-t/\tau_C}, f_D(t) = e^{-t/\tau_D} \quad (2.6)$$

Expression du courant en régime permanent

En faisant tendre le nombre d'itérations vers l'infini dans l'équation 2.5, on obtient le courant en régime permanent pour une largeur d'impulsion et une fréquence donnée :

$$I_{\infty} = \tilde{I} \cdot \frac{e^{-\beta \cdot \Delta_{\max}} \cdot (1 - f_C(W)) \cdot f_D(T - W) + 1 - f_D(T - W)}{1 - f_C(W) \cdot f_D(T - W)} \quad (2.7)$$

Si $T = W$, alors $f_D(T - W) = f_D(0) = 1$ et les NPs sont chargées continuellement. Lorsque la capacité maximale de charge des NPs est atteinte, $I_{\infty} = \tilde{I} \cdot e^{-\beta \cdot \Delta_{\max}}$. Au contraire, si $T \gg W$, les NPs sont complètement relaxées entre chaque impulsion et $I_{\infty} = \tilde{I}$. La figure 2.4 compare le courant en régime permanent théorique avec les mesures expérimentales, pour différentes fréquences d'impulsions. Les deux courbes concordent bien.

2.2.3 Extraction des paramètres du modèle

Notre modèle intègre les quatre paramètres physiques introduit précédemment : les constantes de temps de charge et de décharge des NPs τ_C et τ_D , la variation maximale de l'énergie de Fermi Δ_{\max} et \tilde{I} , qui dépend principalement de la polarisation du dispositif. Le tableau 2.1 liste les paramètres utilisés pour la fonction d'ajustement («fitting» de courbes), avec ces quatre paramètres. L'ordre de grandeur et les contraintes

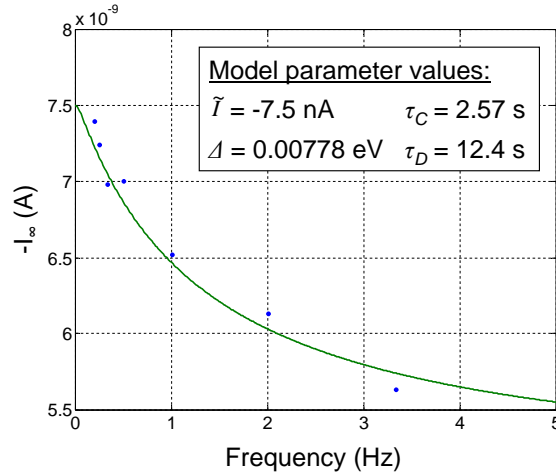


FIGURE 2.4 – Courant en régime permanent calculé à partir du modèle (courbe verte) et issu des mesures expérimentales (points bleus). L’ajustement de courbe est basé sur plusieurs séquences de fréquence constante. La largeur de l’intervalle inter-électrode du NOMFET est de $12 \mu\text{m}$ et les NPs font 5 nm de diamètre. W est fixé à 200 ms et T varie entre 200 ms et l’infini. Le courant en régime permanent est de -5.6 nA lorsque les NPs sont entièrement chargées, contre -7.5 nA lorsqu’elles sont déchargées.

physiques/mathématiques sont spécifiés pour chaque paramètre. La valeur initiale des paramètres pour la première itération de l’ajustement est choisie en accord avec ces ordres de grandeur, qui peuvent également être utilisés pour fixer des limites hautes et basses et améliorer la convergence lorsque c’est nécessaire.

TABLE 2.1 – Paramètres présents dans la fonction d’ajustement.

Paramètre	Ordre de grandeur	Contrainte	Dépendance
\tilde{I}	$1 \mu\text{A} - 1 \text{ nA}$	< 0	Polarisation, déposition du pentacène
Δ_{max}	$0.001 \text{ eV} - 0.01 \text{ eV}$	> 0	Distribution des NPs
τ_C	$1 - 10 \text{ s}$	> 0	Distribution des NPs, largeur du canal
τ_D	$1 - 20 \text{ s}$	> 0	Distribution des NPs, largeur du canal
W	$50 \text{ ms} - 200 \text{ ms}$	$0 < W < T$	(signal d’entrée)
T	$0.2 \text{ s} - 20 \text{ s}$	$W < T < \infty$	(signal d’entrée)
I_0	Même que \tilde{I}	< 0	Aucune (courant initial)

Plusieurs mesures de la réponse en courant pour des séquences d’impulsions de tension avec une fréquence variable ont été utilisées pour l’ajustement. L’outil «nonlinear curve fit» d’OriginPro 8.0 et le langage OriginC ont été utilisés. Celui-ci permet d’utiliser des variables statiques directement dans la fonction itérative (voir 2.5). Parce qu’il y a moins de données expérimentales pour les trains d’impulsions de basse fréquence que pour les hautes fréquences durant le même intervalle de temps, un poids inversement proportionnel à la fréquence des impulsions est appliqué à chaque donnée. Cela assure que la même importance est donnée à chaque partie de la séquence d’impulsions pour l’ajustement, quelle que soit la fréquence. En suivant cette méthodologie, les figures 2.5 et 2.6 montrent la superposition des données expérimentales avec le modèle pour une séquence d’impulsions particulière, pour deux dispositifs différents avec un intervalle inter-électrode de $12 \mu\text{m}$ et de 200 nm respectivement. La diminution de l’intervalle inter-électrode permet des tensions de fonctionnement plus faibles, compatibles avec des circuits «Complementary Metal Oxide Semiconductor» (CMOS). Un autre point important à noter est la réduction de la constante de temps τ_C à une valeur du même ordre que celle typiquement observée dans les synapses biologiques (Abbott et al., 1997).

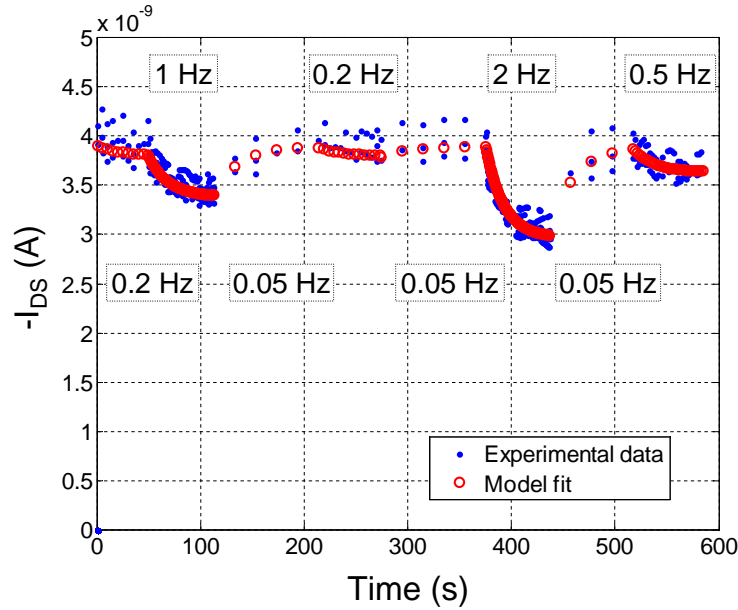


FIGURE 2.5 – Comportement du NOMFET avec une séquence d’impulsions incluant différentes fréquences. Les cercles rouges correspondent au modèle itératif. La valeur des paramètres extraits de cet ajustement est donnée sur la figure 2.9. La largeur de canal du NOMFET est de $12 \mu\text{m}$ et la taille des NPs est de 5 nm .

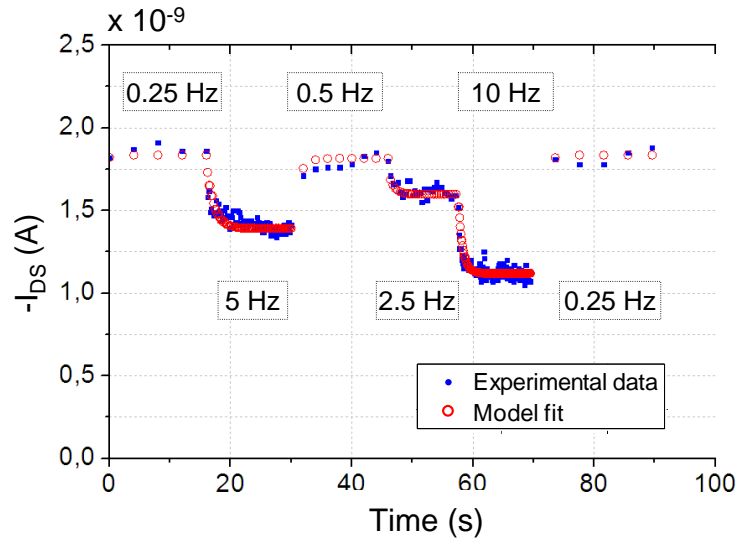


FIGURE 2.6 – Ajustement de courbe avec un NOMFET de largeur de canal 200 nm et des impulsions d’amplitude -3 V . La durée des impulsions est de 50 ms et la taille des NPs est de 10 nm . $\bar{I} = 1.84 \text{ nA}$, $\Delta_{\text{max}} = 0.0203 \text{ eV}$, $\tau_C = 0.387 \text{ s}$ et $\tau_D = 1.08 \text{ s}$.

Le processus de fabrication par auto-assemblage du NOMFET et notamment les variations dans le dépôt des NPs peut conduire à une variabilité importante, qui peut avoir un impact considérable dans la conception d’architectures et de circuits. C’est pourquoi la variabilité doit être prise en compte et intégrée dans le modèle fonctionnel. La figure 2.7 présente le résultat de la modélisation statistique pour un ensemble de 7 dispositifs. Une loi normale est utilisée pour modéliser la distribution statistique des paramètres. Avec ces données, la dispersion obtenue est assez pessimiste, puisque des valeurs non physiques sont statistiquement possibles pour certains paramètres, notamment le courant drain-source. Dans l’implémentation du modèle, les paramètres ne peuvent pas prendre des valeurs au-delà de leurs limites physiques. Par exemple, \bar{I} est

forcé à zéro si sa valeur initiale est positive, émulant ainsi un dispositif non fonctionnel. L'apprentissage neuromorphique post-fabrication, intrinsèquement tolérant aux variations et aux dispositifs défectueux, devrait permettre à une telle architecture d'ignorer ces dispositifs non fonctionnels.

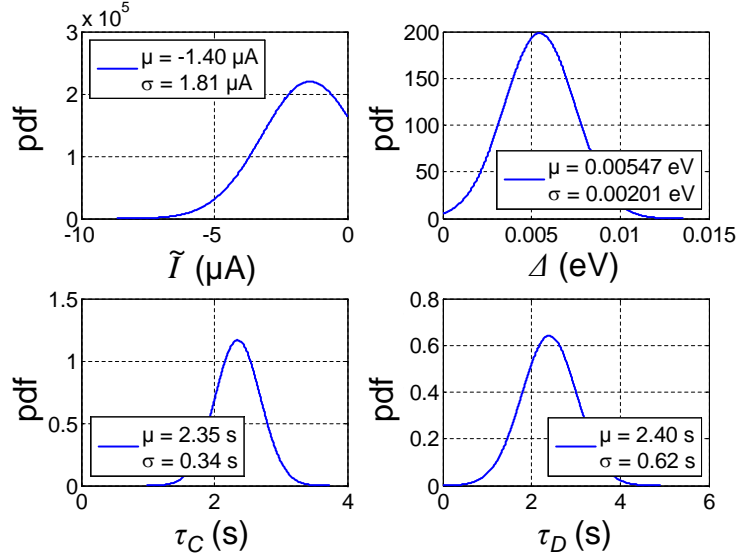


FIGURE 2.7 – Modélisation statistique des quatre paramètres physiques du modèle. «pdf» signifie fonction de densité de probabilité. Ces statistiques sont obtenues après ajustement du modèle avec 7 dispositifs (largeur du canal $5 \mu\text{m}$, taille des NPs 5 nm). La valeur moyenne μ et l'écart-type σ sont calculés pour chaque paramètre à partir des 7 caractérisations expérimentales et la loi normale est utilisée.

2.2.4 Implémentation et simulations

Le modèle électrique itératif a été implémenté dans la langage de description matériel Verilog-A (Agi, 2007), qui peut être utilisé directement dans le simulateur de circuit Spectre de Cadence. Il permet au concepteur de prédire les performances du NOMFET dans des circuits de test et de simuler des circuits hybrides avec d'autres composants (par exemple transistors MOS). Dans notre implémentation, le courant I_{DS} est calculé au début de chaque impulsion. Il dépend de W et T , qui sont obtenus en mesurant les fronts descendant et montant de chaque impulsion durant la simulation : $T = t_{n+1} - t_n$ et $W = t_{n+} - t_n$ (voir la figure 2.3). Les autres paramètres sont extraits de l'ajustement de courbe. Les valeurs des paramètres physiques sont initialisées suivant la distribution statistique de la figure 2.7 au pas initial de la simulation, de manière indépendante pour chaque NOMFET. C'est indispensable pour simuler la robustesse des architectures de circuit. Plusieurs fonctions analogiques et numériques intégrées dans Verilog-A ont été utilisées pour améliorer l'efficacité de la simulation, par exemple, la fonction aléatoire `$rdist_normal` ou la fonction de détection de front `@(cross)...`

Une simulation transitoire Spectre et les mesures expérimentales correspondantes, pour le même train d'impulsions, sont présentées figure 2.10, qui montre une bonne correspondance dans le comportement dynamique. Afin de montrer la précision et les capacités de prédiction du modèle, la séquence d'entrée utilisée pour la comparaison est différente de celle utilisée pour l'ajustement des paramètres du modèle, figure 2.5. Les comportements de dépression et de facilitation synaptique sont observés, ceux-ci dépendant à la fois de la fréquence d'entrée et de l'historique des niveaux de courant. La séquence de 1 Hz est un exemple démontrant ce dernier point (voir figure 2.10) : le poids ou la conductance du NOMFET peut augmenter ou diminuer en fonction de son

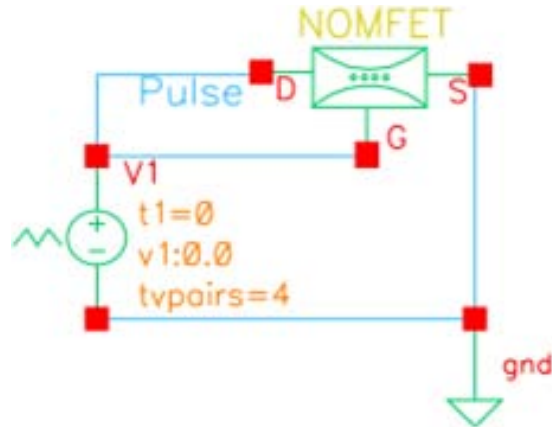


FIGURE 2.8 – Symbole du NOMFET et circuit de test dans l’éditeur de schéma de Cadence. Le symbole a trois terminaux comme le dispositif réel. Dans notre implémentation, les terminaux de drain et de grille sont connectés ensemble pour obtenir un comportement synaptique.

état précédent. Le circuit de test pour le NOMFET, simulé sur la plateforme conception assistée par ordinateur (CAO) de Cadence 5.1.41 (Cad, 2008a) est montré figure 2.8. La séquence d’impulsions de tension d’entrée est générée avec une source de tension linéaire par morceaux. Un symbole est créé pour le NOMFET pour faciliter la conception de circuits hybrides et hiérarchiques avec de la technologie CMOS.

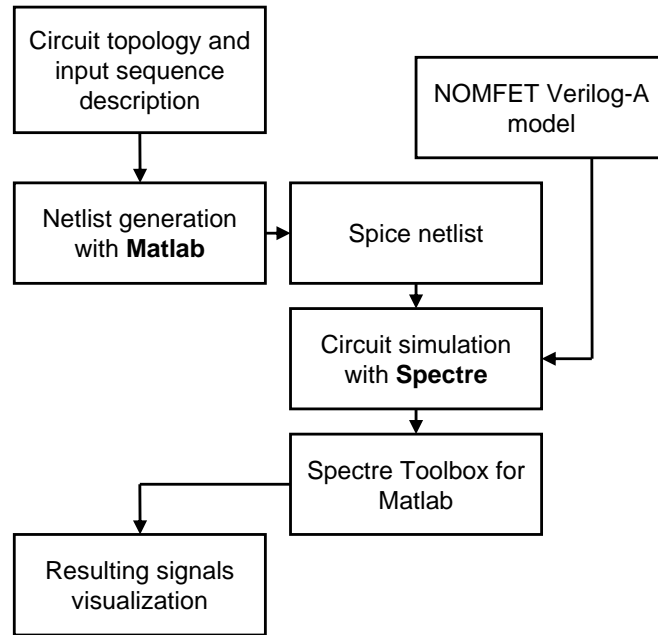


FIGURE 2.9 – Flot de développement pour la co-simulation d’architectures hybrides nano/CMOS avec Matlab et Spectre de Cadence.

Nous avons développé un flot de simulation spécifique pour le modèle fonctionnel du NOMFET, présenté figure 2.9, différent des méthodes classiques pour la modélisation des dispositifs semi-conducteurs. Nous utilisons une technique de co-simulation Matlab/Cadence Spectre pour gérer les signaux d’entrée/sortie qui peuvent être complexes, avec des fréquences variables. Matlab permet de générer des netlists de circuits hybrides nano/CMOS et de traiter les résultats de simulation. Ainsi, nous bénéficions à la fois des avantages du simulateur Spectre avec son interface Verilog-A et des capacités d’analyse et de génération de signaux puissantes de Matlab. L’outil Spectre Toolbox

for Matlab inclus dans MMSIM (Cad, 2008b) fournit l'interface nécessaire pour importer les résultats de simulation dans Matlab. Ce flot de simulation peut être aisément étendu pour étudier le comportement dynamique d'autres nano-dispositifs synaptiques tels que des transistors synaptiques (Lai et al., 2008; Agnus et al., 2010) ainsi que d'autres dispositifs memristifs (Yang et al., 2008; Li et al., 2008).

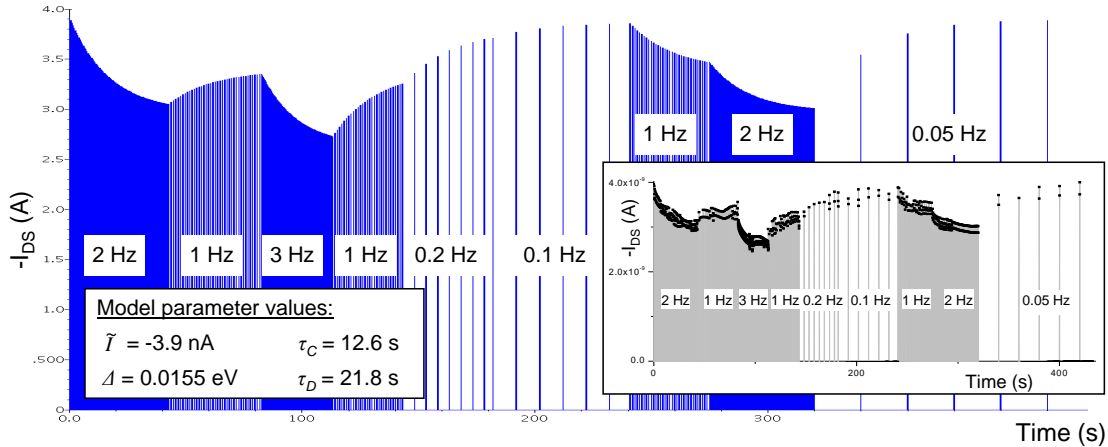


FIGURE 2.10 – La simulation transitoire avec le modèle fonctionnel (courbe bleue) est en accord avec la mesure expérimentale de la réponse en courant drain-source du NOMFET (largeur du canal $12 \mu\text{m}$, taille des NPs 5 nm) (courbe noire). La même séquence d'impulsions est utilisée pour les deux courbes, avec une fréquence variable allant de 0.05 Hz à 3 Hz . En fonction de l'historique du niveau de courant, un comportement facilitateur ou dépressif peut être observé pour une même fréquence (voir par exemple à 1 Hz).

2.2.5 Dépression synaptique et contrôle du gain

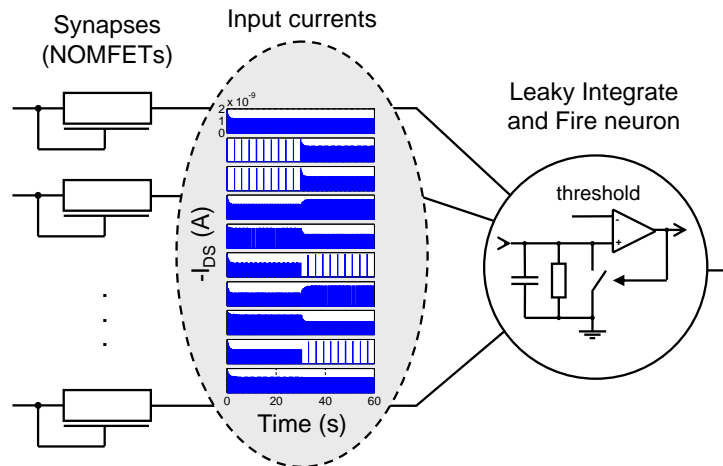


FIGURE 2.11 – Circuit utilisé pour simuler la détection d'un changement synchrone et non corrélé du taux d'activité en entrée de 10 NOMFETs, avec une fréquence d'entrée moyenne de 6 Hz . La fréquence de chaque entrée est choisie aléatoirement à l'instant $t = 0 \text{ s}$ et les fréquences sont permutées aléatoirement à $t = 30 \text{ s}$ pour assurer que la fréquence moyenne et la somme des courants restent rigoureusement identiques après la transition. Le seuil du neurone est fixé à $3.65 \mu\text{V}$ et sa constante de temps d'intégration est de 4 s . Les paramètres du modèle du NOMFET sont ceux de la figure 2.6. La variabilité des dispositifs n'est pas prise en compte dans cette simulation.

Le modèle fonctionnel du NOMFET permet de simuler et d'explorer certaines fonctions neuromorphiques complexes. Dans cette partie, nous présentons un exemple

d'étude de dépression synaptique et de contrôle du gain.

Le contrôle transitoire du gain synaptique est un processus biologique dans lequel une dépression synaptique à court terme a lieu pour compenser une hausse de fréquence sur les entrées du neurone. Ce processus peut être mis en évidence avec une simulation basée sur des mesures biologiques (Abbott et al., 1997). Il permet aux neurones d'être sensibles aux changements relatifs de fréquence sur leurs afférents, réduisant ainsi l'effet des fluctuations sur les entrées de haute fréquence, qui peuvent occulter des changements informatifs sur des entrées de plus faible fréquence. Les simulations (figure 2.10) montrent que le poids synaptique (ou la conductivité du canal) du NOMFET diminue lorsque la fréquence augmente, ouvrant la voie à une forme de contrôle du gain. Selon les paramètres du neurone et la plage de fréquence, une sensibilité aux variations relatives de fréquence sur les entrées synaptiques est possible.

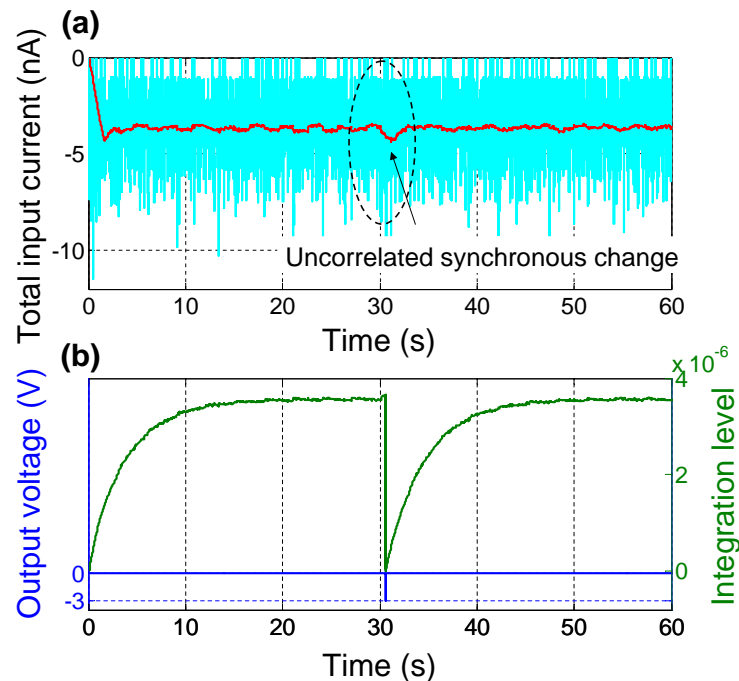


FIGURE 2.12 – (a) Somme des courants provenant des NOMFETs vue à l'entrée du neurone LIF (lissée en rouge), la fréquence des entrées change à 30 s, la fréquence moyenne restant la même (6 Hz). (b) Intégration du neurone (courbe verte) et activité en sortie (courbe bleue). En utilisant la dépression synaptique issue du modèle du NOMFET, la simulation montre que le changement de fréquence peut être détecté à environ 30.5 s.

Une application potentielle du mécanisme de contrôle du gain est la détection de changements synchrones et non corrélés sur le taux d'activité des entrées. Nous reproduisons dans ce qui suit une simulation mettant en évidence ce principe (Abbott et al., 1997) en utilisant le modèle fonctionnel du NOMFET et un modèle Verilog-A de neurone «Leaky integrate-and-fire» (LIF) (Stein, 1967; Tuckwell, 1988). Un neurone associé à des milliers de synapses est avant tout sensible à la fréquence moyenne de ses afférents. Il n'est pas possible, sans dépression synaptique, de détecter un changement synchrone et non corrélé dans le taux d'activité des entrées si la fréquence moyenne ne change pas. Au contraire, avec de la dépression synaptique, le poids synaptique est réduit pour les hautes fréquences. Ainsi, lorsqu'un changement du taux d'activité des afférents se produit, les synapses subissant une augmentation brutale de leur activité auraient une contribution temporairement plus importante à l'intégration du neurone, avant que leur poids ne se réduise par contrôle du gain synaptique. Ce mécanisme est illustré sur la figure 2.11, aux transitions de fréquence basse vers haute. Même si la

fréquence moyenne reste inchangée, l'effet du contrôle du gain peut être suffisant pour déclencher le neurone, comme cela est montré par la simulation figure 2.12. Le seuil du neurone est fixé à $3.65 \mu\text{V}$ et sa constante de temps d'intégration est de 4 s. En accord avec (Abbott et al., 1997), la variabilité des dispositifs n'est pas incluse dans cette simulation, afin de montrer que le changement de fréquence peut être détecté uniquement avec la plasticité du NOMFET et non grâce à la dispersion.

2.3 Mémoire à changement de phase (PCM, PRAM ou PCRAM)

La mémoire à changement de phase est parmi les différentes technologies memristives l'une des plus matures (Wuttig and Yamada, 2007). Les matériaux à changement de phase sont étudiés depuis les années 1960 et sont largement utilisés dans les supports de stockage optiques. Dans ces supports, c'est la modification de la réflectivité provoquée par le changement de phase qui permet d'encoder l'état d'un bit. Mais le changement de phase s'accompagne également d'une modification de la résistance électrique du matériau. En phase cristalline, le matériau est fortement conducteur et fortement réflecteur (état ON, ou SET), tandis qu'en phase amorphe, c'est l'inverse (état OFF, ou RESET). De nombreuses entreprises cherchent à l'heure actuelle à développer des produits mémoire non volatile basés sur la PCM, avec pour principal objectif d'être complémentaire ou supplanter la mémoire flash. On citera Micron (anciennement Numonyx) (Bedeschi et al., 2009), IBM (Nirschl et al., 2007) et Samsung Electronics (Lee et al., 2007b), qui a récemment réalisé une puce mémoire atteignant 1 Go (Chung et al., 2011). Des dispositifs PCM en technologie 45 nm ont également déjà été produits avec succès (Servalli, 2009).

Ces travaux ont été publiés en collaboration avec le CEA LETI dans (Suri et al., 2012) (Suri et al., 2011) (Suri et al., 2012).

2.3.1 Description

Le mécanisme de commutation de la PCM est induit par le changement de phase réversible d'un verre de chalcogénure. Celui-ci peut passer d'une phase amorphe (forte résistance) à une phase cristalline (faible résistance) et inversement par effet Joule. La figure 2.13 montre la structure physique d'une cellule PCM. Pour une intégration dans une mémoire classique, celle-ci est accompagnée d'un transistor de sélection. Le matériau à changement de phase est un chalcogénure, inséré entre deux électrodes de métal et un plot en tungstène formant une résistance chauffante. Deux types de chalcogénure ont été caractérisés par le CEA LETI dans la suite : $\text{Ge}_2\text{Te}_2\text{Sb}_5$ (GST) et GeTe.

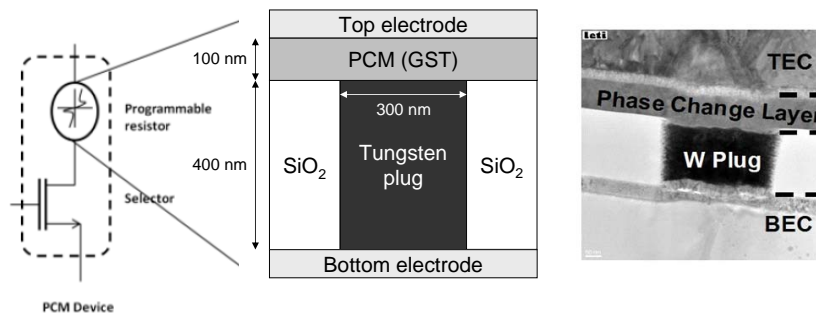


FIGURE 2.13 – Structure physique d'un dispositif PCM. A droite, une image obtenue au microscope électronique à balayage – «Scanning Electron Microscopy» (SEM) du CEA LETI.

Lorsqu'une tension est appliquée aux bornes des électrodes de métal, un courant traverse le plot en tungstène et la couche de chalcogénure. Pour que la PCM passe d'un état cristallin à un état amorphe, la température doit dépasser sa température de fusion, de l'ordre de 600°C pour du GST, ce qui nécessite une impulsion en courant de plusieurs dizaines de nanosecondes. La chute de température doit ensuite être suffisamment rapide pour que le matériau n'ait pas le temps de cristalliser. Le passage du courant doit donc être interrompu rapidement, en quelques nanosecondes. Pour ramener le matériau dans un état cristallin, la température doit être supérieure à la température de transition vitreuse, mais rester inférieure à celle de fusion et cela suffisamment longtemps pour que la cristallisation ait le temps de se faire (durée de l'ordre de 50 à 150 ns pour une cristallisation complète) (Peng et al., 1997). La matrice cristalline correspond à la configuration structurale de niveau d'énergie minimal (Wuttig and Yamada, 2007) et se restaure avec l'énergie thermique. Les profils de température pour l'amorphisation et la cristallisation sont représentés sur la figure 2.14. La caractéristique R-I de la figure 2.15 montre l'évolution progressive de la résistance du matériau en fonction de la densité de courant appliquée.

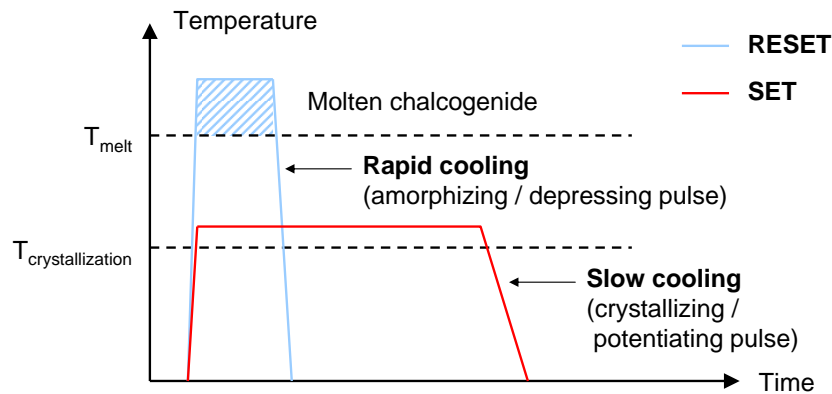


FIGURE 2.14 – Profils de température pour l'amorphisation et la cristallisation (Suri et al., 2011).

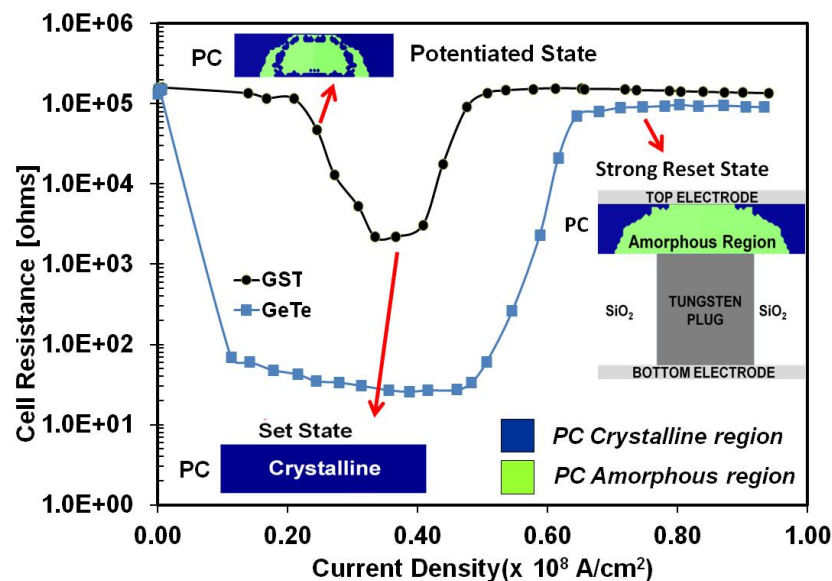


FIGURE 2.15 – Caractéristique R-I des dispositifs GST et GeTe, avec une représentation de la répartition des phases du matériau selon l'état du dispositif (caractérisation du CEA LETI).

2.3.2 Caractérisation expérimentale

La cristallisation étant un processus progressif, il est possible avec des impulsions courtes de ne cristalliser que partiellement le matériau. Ce processus est en plus cumulatif, c'est-à-dire que le taux de cristallisation peut être augmenté graduellement par une succession d'impulsions de programmation. Ce mécanisme a pour effet d'augmenter progressivement la conductance de la PCM, comme le montre les caractérisations expérimentales de la figure 2.16, pour les dispositifs d'IBM et des figures 2.17 et 2.18, pour les dispositifs du CEA LETI. Cela correspond du point de vue d'une synapse à une augmentation de son poids, ou une potentialisation à long terme – «Long-Term Potentiation» (LTP). Il est également possible d'obtenir des niveaux de conductance intermédiaire via l'amorphisation pour émuler une dépression à long terme – «Long-Term Depression» (LTD), comme le montre la caractéristique R-I de la figure 2.15, mais de manière non cumulative.

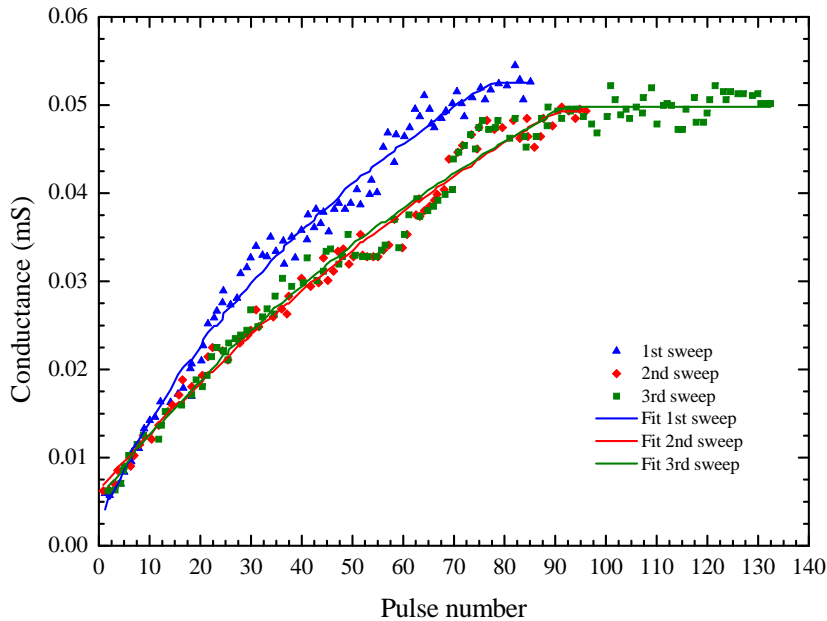


FIGURE 2.16 – Caractéristiques expérimentales de LTP pour les dispositifs PCM GST d'IBM (Breitwisch et al., 2010). Celle-ci est obtenues avec des impulsions de courant de 0.2 mA et de 5 ns.

2.3.3 Modélisation

Modèle phénoménologique

Afin de pouvoir modéliser des millions de synapses PCM dans des réseaux de neurones à grande échelle et ainsi pouvoir évaluer le potentiel d'une technologie synaptique PCM, une modélisation efficace de son comportement est hautement désirable. Dans ce but, nous avons introduit le modèle phénoménologique suivant pour modéliser les caractéristiques de LTP des dispositifs GST et GeTe :

$$\frac{dG}{dt} = \alpha \cdot \exp\left(-\beta \cdot \frac{G - G_{min}}{G_{max} - G_{min}}\right) \quad (2.8)$$

où G est la conductance du dispositif et α , β sont des paramètres d'ajustement. G_{min} et G_{max} sont respectivement les valeurs minimale et maximale de la conductance du dispositif. Pour modéliser le changement de conductance ΔG après l'application

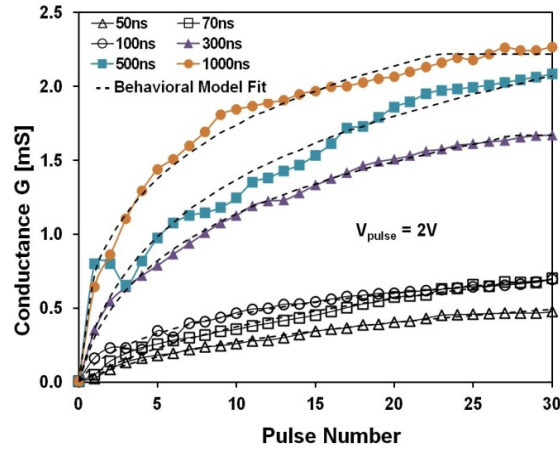


FIGURE 2.17 – Caractéristiques expérimentales de LTP pour les dispositifs PCM GST. Pour chaque courbe, une impulsion de RESET (7 V, 100 ns) est d’abord appliquée, suivie de 30 impulsions identiques consécutives de potentialisation (2 V). Les courbes en pointillés montrent l’ajustement du modèle phénoménologique décrit à l’équation 2.9.

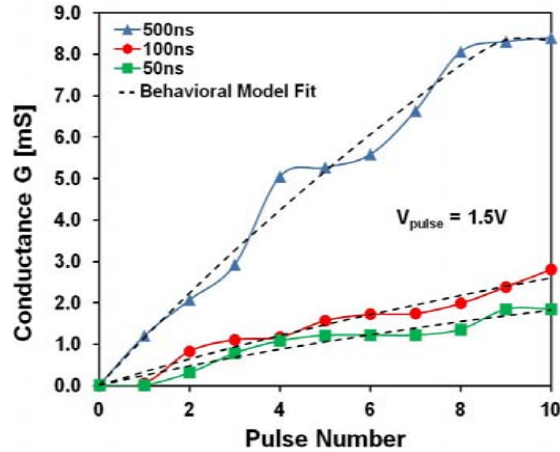


FIGURE 2.18 – Caractéristiques expérimentales de LTP pour les dispositifs PCM GeTe. Les courbes en pointillés montrent l’ajustement du modèle phénoménologique décrit à l’équation 2.9.

d’une courte impulsion de LTP de durée Δt , l’équation précédente peut être intégrée comme suit :

$$\Delta G = \alpha \cdot \Delta t \cdot \exp\left(-\beta \cdot \frac{G - G_{min}}{G_{max} - G_{min}}\right) \quad (2.9)$$

Ces équations modélisent plutôt bien les mesures faites à la fois sur les dispositifs GST et GeTe, comme illustré sur les figures 2.17 et 2.18. Ceci est important car la forme de la courbe de potentialisation peut avoir un impact important sur les performances du système comme le suggèrent les travaux en neurosciences computationnelles (Rubin et al., 2001). Pour les dispositifs GST et GeTe, nous avons utilisé un unique jeu de paramètres α , β , G_{min} et G_{max} pour différentes largeurs d’impulsions. Le tableau 2.2 liste les valeurs des paramètres pour l’ajustement de courbes avec les données GST et GeTe des figures 2.17 et 2.18.

TABLE 2.2 – Paramètres pour le modèle phénoménologique des courbes de LTP des dispositifs GST et GeTe.

Paramètre	GST (300 ns)	GeTe (100 ns)
G_{min} (μ S)	8.50	8.33
G_{max} (mS)	2.3	2.9
α (S/s)	1100	3300
β	-3.8	-0.55

Modèle compatible circuit

Afin de pouvoir concevoir un circuit neuronal hybride composé de neurones CMOS et de synapses PCM, un modèle compatible circuit pour la PCM est nécessaire. Nous avons développé un tel modèle, qui est spécifiquement conçu pour reproduire correctement le comportement progressif des expériences de LTP présentées précédemment. Ce modèle circuit simple, inspiré de (Sonoda et al., 2008), a été développé en VHDL-AMS et inclus les comportements de LTP et de LTD. Les simulations ont été faites à l'aide du simulateur mixte analogique/numérique – «Analog and Mixed-Signal» (AMS) de Cadence. La figure 2.20 montre les courbes de LTP simulées pour six largeurs d'impulsion différentes pour un dispositif GST. Le tableau 2.3 liste l'ensemble des constantes et des paramètres d'ajustement utilisés dans ce modèle circuit. Le modèle se décompose en trois parties : électrique, thermique et changement de phase. Pour la partie électrique, la relation entre le courant et la tension est supposée ohmique :

$$v = R_{gst} \cdot i \quad (2.10)$$

Nous avons préféré ne pas inclure la non-linéarité du courant à forte tension («threshold switching effect») dans ce modèle, car il est avant tout conçu pour émuler le comportement de LTP durant l'apprentissage dans des réseaux de neurones de grande taille, où la vitesse de simulation est essentielle. R_{gst} est la résistance faible champ du dispositif, qui consiste en la somme de la résistance de la couche GST et de la résistance des électrodes du dessous et du dessus R_s . La résistance de la couche à changement de phase est une fonction de la fraction du volume en phase amorphe C_a :

$$R_{gst} = R_s + R_{0c}^{(1-C_a)} \cdot R_{0a}^{C_a} \quad (2.11)$$

R_{0c} et R_{0a} correspondent respectivement aux résistances de l'état entièrement cristallisé et entièrement amorphisé. Nous avons utilisé une interpolation logarithmique, qui est intermédiaire entre les cas série et parallèle (Sonoda et al., 2008), ce qui conduit au meilleur résultat pour nos dispositifs GST.

La technologie PCM est affectée par le phénomène de dérive de la résistance («resistance drift»). Des relaxations structurelles dans le matériau en phase amorphe provoquent la dérive de sa résistance dans le temps (voir figure 2.19). Cela se produit à température ambiante, jusqu'à ce que la résistance atteigne une valeur de saturation (Karpov et al., 2007). Cet effet est d'autant plus marqué que la fraction du volume en phase amorphe C_a , et donc la résistance de la PCM, est grande. Afin d'évaluer l'impact de la dérive de la résistance sur la stabilité de l'apprentissage, nous avons inclus une modélisation comportementale de ce phénomène, qui peut être activé ou non dans la simulation. Pour ce faire, R_{0a} est remplacé par R_a :

$$R_a = R_{0a} + \left(\frac{t - t_0}{t_d} \right)^{C_a \cdot dr} \quad (2.12)$$

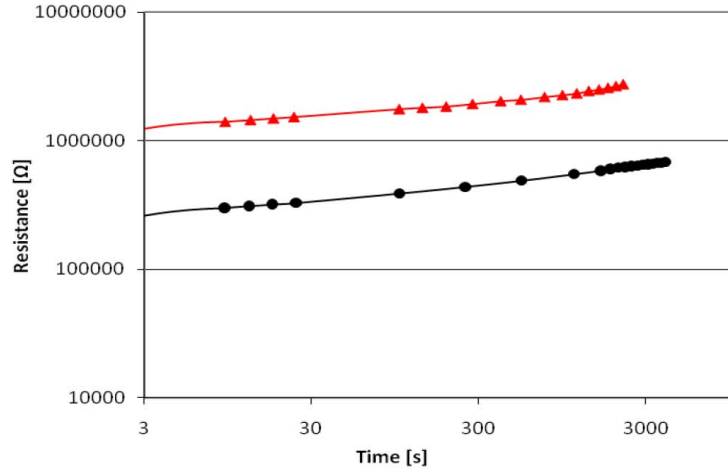


FIGURE 2.19 – Mesure expérimentale de la dérive de la résistance PCM dans le temps (Suri et al., 2011). La première caractéristique (ronds) est obtenue pour une tension de programmation initiale de 8 V, tandis que la seconde (triangles), pour une tension initiale de 10 V.

où t_0 est l'instant auquel le dernier changement de phase a eu lieu (c'est-à-dire la dernière fois où T_b a atteint T_g).

Dans la partie thermique du modèle, la puissance électrique P_t et la température du matériau à changement de phase T_b sont liés par l'équation suivante, avec T_0 la température ambiante :

$$T_b = T_0 + P_t \cdot R_{tgst} \quad (2.13)$$

La résistance thermique du matériau à changement de phase R_{tgst} est décrite par l'équation :

$$R_{tgst} = (1 - C_a) \cdot R_{tc0} + C_a \cdot R_{ta0} \quad (2.14)$$

où R_{tc0} et R_{ta0} sont les résistances thermiques de l'état entièrement cristallisé et entièrement amorphisé respectivement. Avec l'augmentation de la tension aux bornes du dispositif, le courant dans la phase amorphe subit une non-linéarité. A forte tension, la résistance de la phase amorphe se rapproche de celle de la phase cristalline. La puissance électrique durant la change de phase est alors essentiellement indépendante du taux de phase amorphe. La puissance électrique est donc calculée en utilisant la résistance équivalente à celle du matériau entièrement cristallisé, au lieu de la résistance faible champ :

$$P_t = \frac{v^2}{R_s + R_{0c}} \quad (2.15)$$

La partie du modèle changement de phase utilise des équations comportementales. L'amorphisation a lieu lorsque la température T_b est supérieure à T_m , la température de fusion. La vitesse d'amorphisation est supposée augmenter linéairement avec la température et est nulle lorsque $T_b = T_m$, assurant ainsi la continuité avec la vitesse de cristallisation à cette température. Cela donne l'équation :

$$\frac{dC_a}{dt} = \frac{1}{\tau_a} \cdot \frac{T_b - T_m}{T_m} \text{ lorsque } T_b > T_m \quad (2.16)$$

La cristallisation est régie par deux mécanismes, la nucléation et la croissance. La prédominance de l'un sur l'autre dépend du matériau. L'équation modélisant la vitesse de cristallisation n'inclut pas un modèle séparé pour la vitesse de nucléation («nucleation-driven») et un modèle pour la vitesse de croissance («growth-driven»). L'expression 2.16 se rapproche toutefois d'un modèle de croissance. Elle inclut un terme C_a^2 car la vitesse de cristallisation dépend typiquement de la surface de l'interface amorphe-cristalline pour le processus de croissance (et du volume pour le processus de nucléation).

$$\frac{dC_a}{dt} = -\frac{C_a^2}{\tau_c} \cdot \left(1 - \exp\left(E_a \cdot \frac{T_b - T_m}{T_b}\right)\right) \cdot \exp\left(-\frac{E_b}{T_b}\right) \text{ lorsque } T_b < T_m \text{ et } T_b > T_g \quad (2.17)$$

Ce modèle, avec les paramètres listés dans la tableau 2.3 donne de bons résultats pour les courbes de LTP, bien que l'ajustement ne soit pas aussi bon que pour le modèle phénoménologique. Une simulation de l'évolution de la conductance en fonction de la tension appliquée, avec les mêmes paramètres, est montrée figure 2.20. Bien que l'ajustement soit moins précis pour des impulsions plus courtes, toutes les courbes de la figure 2.20 correspondent à un unique jeu de paramètres. Le modèle est donc adapté à de l'exploration rapide et de la conception simplifiée de circuit où les dispositifs PCM sont utilisés pour émuler des millions de synapses. Pour référence, le modèle VHDL-AMS est donné listing 2.1.

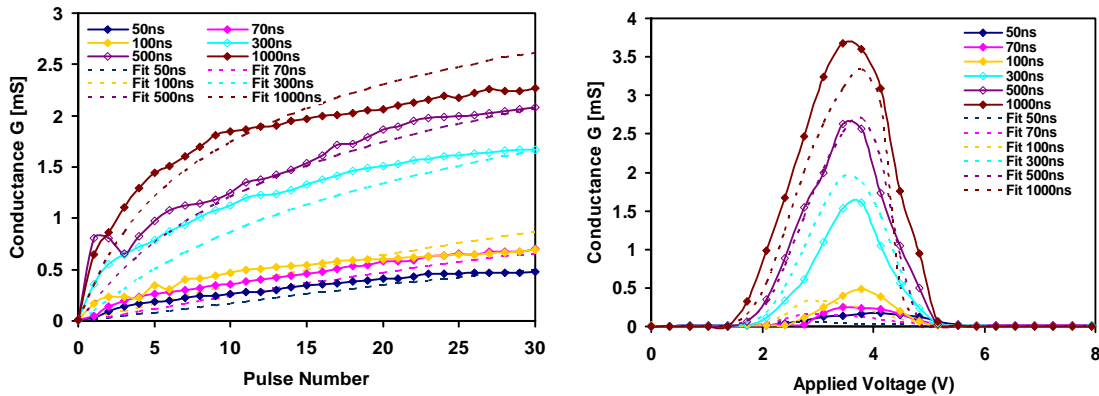


FIGURE 2.20 – Gauche : simulation au niveau circuit des caractéristiques de LTP pour les dispositifs GST. Droite : simulation au niveau circuit de l'évolution de la conductance en fonction de la tension appliquée pour les dispositifs GST avec six largeurs d'impulsion différentes.

Listing 2.1 – Implémentation VHDL-AMS de modèle PCM.

```

1 library IEEE;
2 use IEEE.electrical_systems.all;
3 use IEEE.thermal_systems.all;
4 use IEEE.math_real.all;
5 use IEEE.math_complex.POSITIVE_REAL;
6
7 use WORK.generic_types_pkg.all;
8
9 entity model_pcm is
10     generic(
11         -- Electrical model
12         Rs: RESISTANCE:= 100.0; -- Serial resistance (top and bottom electrodes)
13         ROa: RESISTANCE:= 159.0e3; -- Resistance of the fully amorphized state (ohm)
14         ROc: RESISTANCE:= 135.0; -- Resistance of the fully crystallized state (ohm)
    
```

TABLE 2.3 – Paramètres utilisés pour les simulations de la figure 2.20 avec le modèle semi-compact de la PCM GST.

Paramètre	Valeur	Description
<i>Modèle électrique</i>		
R_s	100 Ω	Résistance série (électrodes d'accès)
R_{0a}	159 k Ω	Résistance de l'état entièrement amorphisé
R_{0c}	135 Ω	Résistance de l'état entièrement cristallisé
<i>Modèle thermique</i>		
R_{ta0}	15.9×10^3 W/K	Résistance thermique de l'état entièrement amorphisé
R_{tc0}	5.07×10^3 W/K	Résistance thermique de l'état entièrement cristallisé
T_0	300 K	Température ambiante
<i>Modèle changement de phase</i>		
E_a	0.335	Paramètre pour la vitesse de cristallisation haute T°
E_b	5.77×10^3	Paramètre pour la vitesse de cristallisation basse T°
T_g	380 K	Température minimale de cristallisation
T_m	698 K	Température de fusion
τ_a	8.17×10^{-13} s $^{-1}$	Taux de cristallisation
τ_c	1.10×10^{-6} s $^{-1}$	Taux d'amorphisation
dr	0.03	Coefficient de dérive
t_d	0.03	Constante de temps de dérive

```

15
16     -- Thermal model
17     Rta0: THERMAL_RESISTANCE:= 15.9e3; -- Thermal resistance of the fully
18           crystallized state (W/K)
19     Rtc0: THERMAL_RESISTANCE:= 5.07e3; -- Thermal resistance of the fully amorphized
20           state (W/K)
21     T0: TEMPERATURE:= 300.0; -- Ambient temperature (K)
22
23     -- Phase-change model
24     Ea: POSITIVE_REAL:= 0.335; -- Fitting parameter for crystallization rate
25     Eb: POSITIVE_REAL:= 5.77e3; -- Fitting parameter for crystallization rate
26     Tm: TEMPERATURE:= 698.0; -- Melting point (K)
27     Tg: TEMPERATURE:= 380.0; -- Lowest temperature at which crystallization can
28           occur (K)
29     tauC: POSITIVE_REAL:= 8.17e-13; -- Crystallization rate (1/s)
30     tauA: POSITIVE_REAL:= 1.10e-6; -- Amorphization rate (1/s)
31     dr: POSITIVE_REAL:= 0.03; -- Drift coefficient
32     td0: POSITIVE_REAL:= 1.0; -- Drift time constant
33
34     -- Behavioral simulation only
35     Ca_init: PERCENT_T:= 0.99 -- Initial amorphous ratio
36 );
37
38 port(
39     terminal p: ELECTRICAL;
40     terminal n: ELECTRICAL
41 );
42 end entity model_pcm;
43
44 architecture behavioral of model_pcm is
45     quantity v across i through p to n;
46
47     -- Electrical model
48     quantity Rgst: RESISTANCE; -- Resistance of the phase-change layer (ohm)
49     quantity Ra: RESISTANCE; -- Resistance of the fully amorphized state, with drift (
50         ohm)

```

```

47
48 -- Thermal model
49 quantity Pt: HEAT_FLOW; -- Thermal power in the phase-change layer (W)
50 quantity Rtgst: THERMAL_RESISTANCE; -- Thermal resistance of the phase-change layer
    (K/W)
51 quantity Tb: TEMPERATURE; -- Temperature of the phase-change layer (K)
52
53 -- Phase-change model
54 quantity Ca: PERCENT_T:= Ca_init; -- Amorphous ratio
55 quantity td: POSITIVE_REAL; -- Drift period (s)
56 begin
57 -- Electrical model --
58 -----
59 Rgst == Rs + (R0c**(1.0 - Ca))*(Ra**Ca);
60 i == v/Rgst;
61
62 -- Drift behavioral modeling
63 Ra == R0a*(((now-td)/td0)**(Ca*dr));
64
65 -- Thermal model --
66 -----
67 Pt == v**2/(Rs + R0c);
68 Rtgst == (1.0 - Ca)*Rtc0 + Ca*Rta0;
69 Tb == T0 + Pt*Rtgst;
70
71 -- Phase-change model --
72 -----
73 if (domain=quiescent_domain) use
74     Ca == Ca_init;
75     td == now;
76 else
77     if (Tb'ABOVE(Tm)) use
78         -- Amorphizing
79         Ca'DOT == tanh(10.0*(1.0 - Ca))*(Tb - Tm)/Tm/tauA;
80         td == now;
81     elsif (Tb'ABOVE(Tg)) use
82         -- Crystallizing
83         Ca'DOT == -(Ca**2)*(1.0 - exp(Ea*(Tb - Tm)/Tb))*exp(-Eb/Tb)/tauC;
84         td == now;
85     else
86         Ca'DOT == 0.0;
87         td'DOT == 0.0;
88     end use;
89 end use;
90
91 break on Tb'ABOVE(Tm);
92 break on Tb'ABOVE(Tg);
93 end architecture behavioral;

```

2.4 Conductive-Bridging RAM (CBRAM)

La CBRAM est également appelée «Programmable-Metallization-Cell» (PMC), ou mémoire à électrolytes solides. Cette technologie est originaire de l'Université d'État de l'Arizona, qui a monté une «start-up» pour valoriser la technologie en 1996, Axon Technologies Corporation (Kozicki and West, 1998). Axon Technologies Corporation a licencié sa technologie à Micron Technology, Inc. en 2002, qui a popularisé la dénomination de CBRAM (Kund et al., 2005), puis à Infineon Technologies AG en 2004. Le brevet original est aujourd'hui cité par plus de 300 brevets, dont les deux tiers ont été déposés par Micron Technology, Inc. Cette technologie est également développée par Altis Semiconductor (initialement une coentreprise entre IBM Microelectronics et

Infineon Technologies), en collaboration avec le CEA. L'une des premières communications portant sur l'utilisation de cette technologie, publiée par son inventeur, porte sur l'apprentissage de Hebb dans un réseau de neurones (Swaroop et al., 1998). Les produits à base de CBRAM tardent à venir sur le marché et aucune puce mémoire de grande capacité n'a pour l'instant été démontrée. Une EEPROM de 1 Mo basée sur de la CBRAM est toutefois commercialisée depuis 2011 par Adesto Technologies en partenariat avec Altis Semiconductor.

Ces travaux ont été publiés en collaboration avec le CEA LETI dans (Suri et al., 2012).

2.4.1 Description

Parmi les matériaux utilisés pour réaliser l'électrolyte solide on trouve du germanium (Ge) combiné avec des chalcogénures (S, Se, Te) (Kozicki et al., 2005), ou du SiO_2 (Schindler et al., 2008). Le principe physique de commutation est la dissolution d'une électrode soluble (Ag, Cu) dans l'électrolyte (migration de cations), ce qui permet de créer ou de détruire un pont conducteur par oxydoréduction, selon la polarité de la tension appliquée au dispositif (voir figure 2.21). La caractéristique statique est donc généralement bipolaire pour ce type de dispositif, souvent avec une asymétrie entre les seuils de SET et de RESET, bien que des caractéristiques unipolaires aient également été observées pour des structures proches (Schindler et al., 2007). Parmi les articles de revue les plus complets sur le sujet on citera (Waser and Aono, 2007) (pour la partie migration de cations). Une limite théorique de la taille d'une cellule a été estimée à 2.5 nm (Zhirnov et al., 2011), mais dépend du mécanisme exact de commutation.

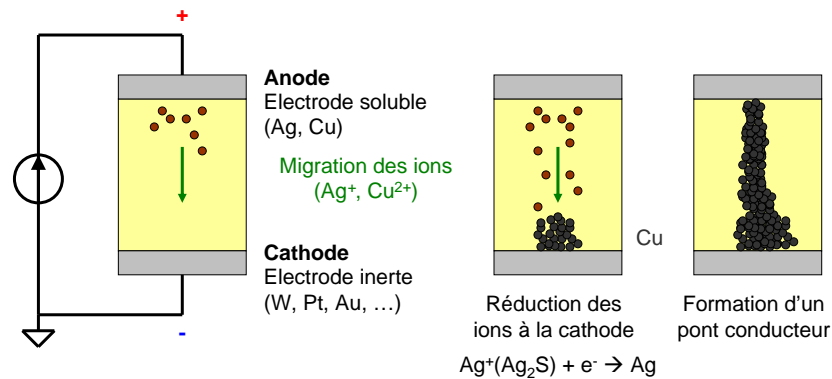


FIGURE 2.21 – Mécanisme de commutation bipolaire avec les électrolytes solides : migration des cations.

La structure des dispositifs CBRAM utilisés dans nos travaux, qui comprend un transistor de sélection pour le contrôle du courant, est schématisée sur la figure 2.22 (structure 1T-1R). La commutation se fait par migration des cations Ag^+ dans la couche GeS_2 , qui se déchargent sur l'électrode inerte en tungstène (W) pour former un filament conducteur (SET). L'inversion du champ électrique provoque la dissolution du filament (RESET). Une commutation binaire sans transistor de sélection est possible, mais la programmation multi-niveaux nécessite de pouvoir contrôler le diamètre du filament conducteur, ce qui se fait en limitant le courant durant la formation de celui-ci (Palma et al., 2012). Ainsi, dans le cadre d'une utilisation en tant que synapse, une LTP progressive est difficile à réaliser. Pour ce faire, il faut adapter la tension de grille du transistor de contrôle au niveau de conductance désiré. Cet effet n'est pas non plus cumulatif, ce qui signifie que pour augmenter graduellement le poids de la synapse, il est nécessaire d'adapter cette tension de contrôle en fonction de l'état précédent de

celle-ci. Une LTD progressive est quant à elle encore plus difficile à obtenir du fait de la difficulté de contrôler la dissolution du filament. Le passage de l'état ON à OFF est abrupt.

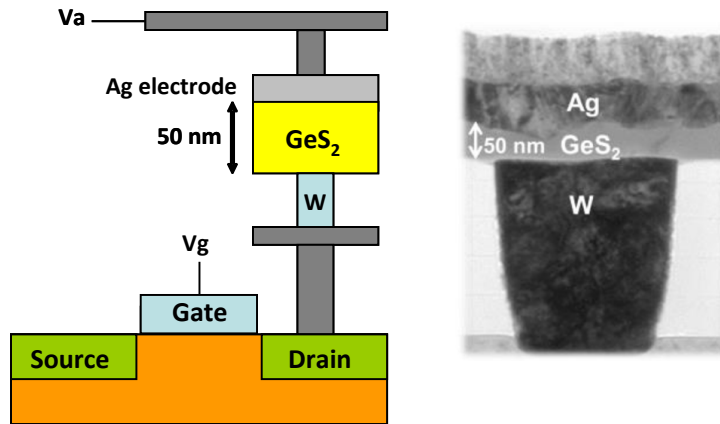


FIGURE 2.22 – Structure d'un dispositif CBRAM avec son transistor de sélection (1T-1R) et image SEM de l'élément CBRAM (à droite).

2.4.2 Caractérisation expérimentale

La caractéristique quasi-statique du dispositif est bien bipolaire, avec une asymétrie sur les seuils de tension de SET et de RESET, comme illustré sur la caractéristique de gauche de la figure 2.23. La caractéristique de droite montre la programmation multi-niveaux de la CBRAM avec son transistor de limitation du courant. Elle est obtenue en augmentant progressivement la tension de grille du transistor, comme illustré dans l'encart de la figure.

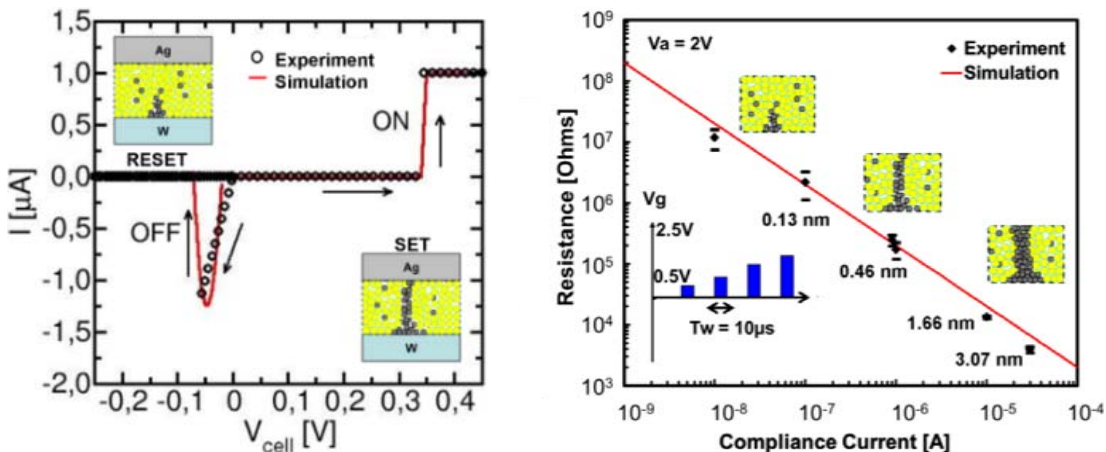


FIGURE 2.23 – Gauche : caractéristique I-V quasi-statique du dispositif CBRAM, mettant en évidence le fonctionnement bipolaire de celui-ci. Droite : programmation multi-niveaux de la CBRAM par contrôle du courant avec le transistor de sélection. Les données simulées sont issues de (Suri et al., 2012) (le rayon du filament conducteur pour les différents niveaux de conductivité est indiqué sur la figure).

La non cumulativité de la CBRAM pour la programmation multi-niveaux ne permet à priori pas de résoudre le «bottleneck» de Von Neumann inhérent à l'apprentissage synaptique. Mais les caractéristiques de cette technologie restent excellentes pour de la programmation binaire, qui reste donc intéressante pour l'implémentation de synapses

binaires. La CBRAM présente en outre une autre caractéristique à la fois très pénalisante pour de la mémoire numérique et potentiellement intéressante pour des synapses : une commutation stochastique intrinsèque. Cette stochasticité pose une contrainte importante sur les conditions de programmation utilisées pour en faire une cellule mémoire. Celles-ci doivent en effet être suffisamment fortes, en termes d'amplitude et de durée de tension de programmation, pour garantir qu'une commutation ait lieu dans 100% des cas. La figure 2.24 montre que la probabilité de commutation diminue lorsque les tensions de SET et de RESET sont réduites. Sur 1000 cycles de commutation, il est par exemple possible de régler la tension de SET pour n'avoir qu'une chance sur deux de passer à l'état ON.

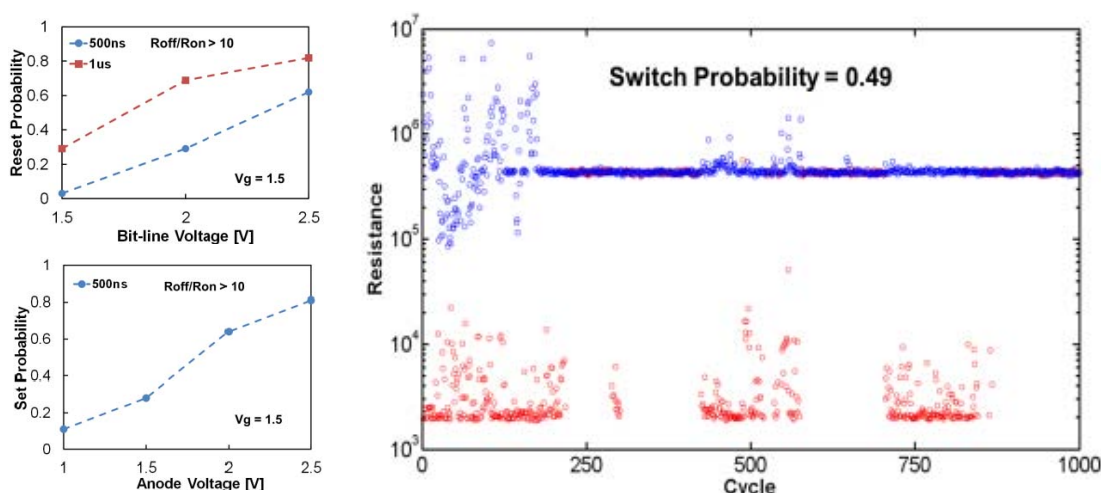


FIGURE 2.24 – Gauche : probabilités de SET et de RESET du dispositif en fonction de la tension de programmation (conditions relâchées par rapport à la programmation binaire). Droite : illustration de la commutation stochastique pour un dispositif CBRAM.

2.5 Resistive RAM (RRAM ou ReRAM)

La RRAM est également appelée OxRRAM, ou mémoire à oxydes métalliques. La CBRAM est parfois incluse dans cette dénomination, tandis que le terme plus générique de « mémoire résistive » comprend généralement les deux. Parmi les articles de revue les plus complets sur le sujet on citera (Lin et al., 2008; Sawa, 2008) pour les oxydes de pérovskite, (Yu et al., 2012) pour les oxydes binaires et (Waser and Aono, 2007) pour la partie migration d'anions. Il reste difficile de parler de la technologie RRAM compte tenu de la multitude de dispositifs et de matériaux différents présentés comme tel. Tout le monde ne s'accorde pas non plus sur les mécanismes de commutation dans ces dispositifs, qui dépendent des matériaux utilisés. Pour ajouter à la confusion, certains de ces dispositifs sont désormais présentés comme des memristors, suite au succès rencontré par la publication de Hewlett-Packard (HP) Labs (Strukov et al., 2008).

2.5.1 Description

Les dispositifs RRAM peuvent être séparés en deux catégories, selon l'oxyde utilisé pour former la couche isolante de la structure métal-isolant-métal (MIM) : oxyde binaire ou oxyde de pérovskite.

Oxyde binaire

L'isolant dans la structure MIM peut être un oxyde binaire métallique (NiO (Baek et al., 2005; Sato et al., 2008), TiO₂, CuO, HfO₂, ZrO₂, ...). La caractéristique statique est généralement unipolaire pour ce type de dispositif, mais elle peut également dans certains cas être bipolaire, pour le même matériau, selon le mécanisme de commutation privilégié (Jeong et al., 2007). L'explication généralement avancée pour expliquer le mécanisme de commutation unipolaire est la formation et la destruction d'un filament conducteur dans l'oxyde via des effets thermiques, comme illustré sur la figure 2.25. La formation se fait par claquage électronique dans le diélectrique tandis que la destruction se fait par effet Joule. Il s'agit d'un mécanisme de type «fuse-antifuse», pour lequel la limitation de courant («current compliance») durant le SET a un impact fort sur la valeur de la résistance ainsi obtenue (Cao et al., 2009). S'il est possible d'obtenir plusieurs niveaux de conductance, ce mécanisme n'est à priori pas le plus favorable à une utilisation en synapse, du fait qu'il ne permet pas facilement d'augmenter ou de diminuer la valeur de la résistance de manière incrémentale (absence d'effet cumulatif). Le mécanisme de commutation pourrait en outre être stochastique (Jo et al., 2009b).

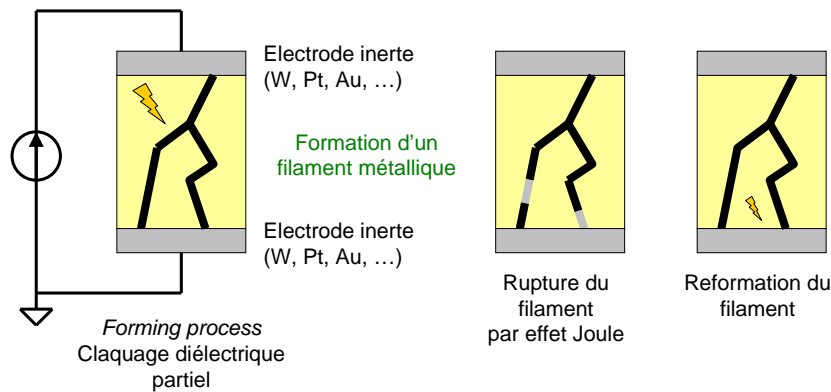


FIGURE 2.25 – Mécanisme de commutation unipolaire pour les oxydes métalliques : formation et destruction d'un filament métallique.

Oxyde de pérovskite

L'isolant dans la structure MIM peut être un oxyde ternaire, à structure de pérovskite, de formule générale ABO₃ (SrZrO₃ (Beck et al., 2000), SrTiO₃ (Szot et al., 2006), ...). L'inconvénient de cette technologie est qu'elle est plus difficile à mettre en œuvre que les oxydes binaires métalliques (Baek et al., 2004). La caractéristique statique est généralement bipolaire pour ce type de dispositif. Le mécanisme physique de commutation le plus souvent reporté pour cette structure est la migration d'anions, plus précisément d'ions oxygène et leur réaction à l'interface entre l'anode et l'oxyde, comme illustré sur la figure 2.26.

Ce mécanisme est également observé avec des oxydes binaires et notamment le TiO₂. Ceci est très certainement le cas du memristor de HP Labs. Celui-ci a une structure MIM Pt/TiO₂/Pt, dans laquelle l'oxyde est composé de deux couches, l'une de TiO₂ et l'autre de TiO_{2-x}, déplié en atome d'oxygène et de plus grande conductivité. Le mécanisme de changement de conduction est attribué à la migration des lacunes d'oxygène, qui a pour effet de déplacer la frontière entre ces deux couches (Strukov et al., 2008) et/ou de créer et détruire des chemins conducteurs dans l'oxyde (Do et al., 2009).

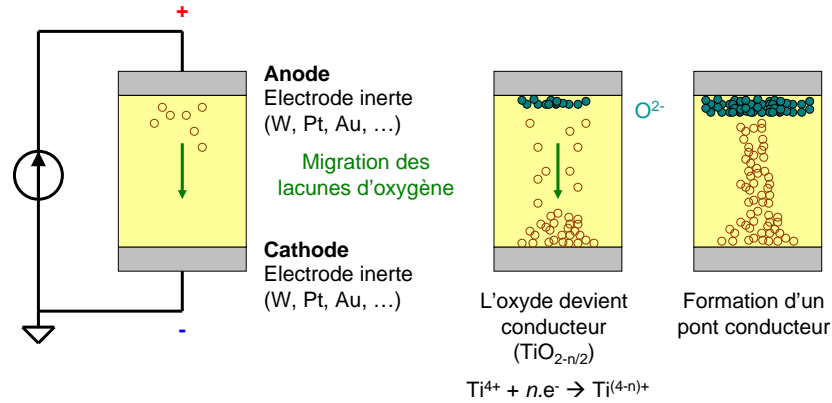


FIGURE 2.26 – Mécanisme de commutation bipolaire pour les oxydes métalliques : migration des anions (ou de manière équivalente, migration des lacunes d'oxygène).

2.5.2 Exemples de caractéristiques

Nous présentons brièvement dans cette partie les caractéristiques, pour de la programmation graduelle, des deux dispositifs les plus populaires pour réaliser des synapses : la nano-synapse de l'Université du Michigan (figure 2.27) et le memristor TiO_2 de HP Labs (figure 2.28). La nano-synapse a été spécialement optimisée pour une utilisation en tant que synapse. Les dispositifs précédents de la même équipe ont un fonctionnement binaire (Jo and Lu, 2008), mais également stochastique (Jo et al., 2009b). Un mélange Ag/Si est pulvérisé dans la couche isolante du dispositif, avec un gradient de concentration pour l'argent. Le champ électrique appliqué aux bornes du dispositif a pour effet de déplacer les atomes d'argent pour donner naissance à une zone riche en Ag et une zone pauvre en Ag, dont l'épaisseur détermine la conductance de la synapse.

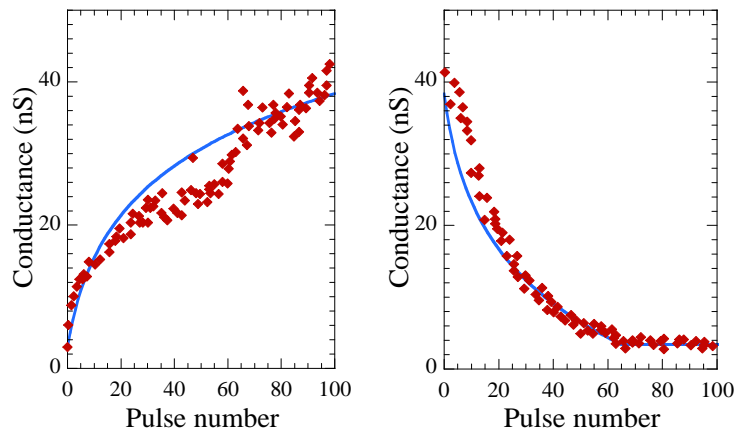


FIGURE 2.27 – Caractéristiques de la nano-synapse de l'Université du Michigan. Gauche : évolution de la conductance (mesurée sous 1 V) pour une série d'impulsions de potentialisation (de -3.2 V). Droite : évolution de la conductance pour une série d'impulsions de dépression (de 2.8 V). La durée des impulsions est de 300 μs . Données expérimentales de (Jo et al., 2010).

L'extraction à partir des données issues de la littérature et la modélisation des caractéristiques présentés ici ont été publiés dans (Querlioz et al., 2011b). La figure 2.27 montre que la conductance de la nano-synapse peut être à la fois augmentée et diminuée de manière progressive et cumulative, puisque les caractéristiques sont obtenues pour une série d'impulsions identiques. Cela fait de ce dispositif un excellent candidat pour implémenter une synapse, qui peut être renforcée ou affaiblie par une simple impulsion

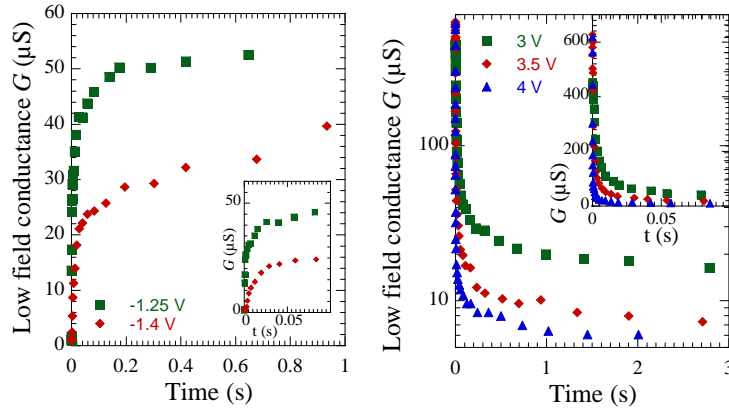


FIGURE 2.28 – Caractéristiques du memristor de HP Labs. Gauche : potentialisation, pour des impulsions de -1.4 V et -1.25 V. Droite : dépression, pour des impulsions de 4 V, 3.5 V et 3 V. Données expérimentales de (Pickett et al., 2009).

de tension, sans nécessiter de connaître son état au préalable. Des caractéristiques similaires sont obtenues pour le memristor, sur la figure 2.28. Les changements de conductance sont toutefois plus brutaux que pour le dispositif de l’Université du Michigan. Notamment, la diminution de la conductance est abrupte (contrairement aux autres caractéristiques, la caractéristique correspondante est tracée en échelle logarithmique).

2.6 Dispositifs à jonction tunnel

Il en existe deux types : les jonctions tunnel magnétiques – «Magnetic Tunnel Junctions» (MTJs), à la base de la MRAM, et les jonctions tunnels ferroélectriques, à la base de la mémoire ferroélectrique (FRAM ou FeRAM). Ces dispositifs entrent également dans la famille des dispositifs memristifs.

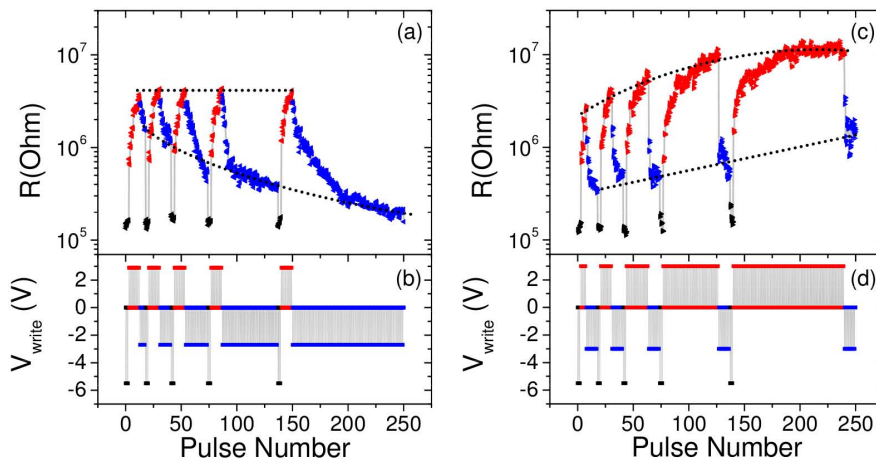


FIGURE 2.29 – Caractéristiques du memristor ferroélectrique de CNRS/Thales, pour une succession d’impulsions de programmation identiques (Chanthbouala et al., 2012). Évolution de la résistance de la jonction avec des séquences d’impulsions de différentes amplitudes (à gauche pour 2.9 V et -2.7 V et à droite pour 3 V et -3 V).

Les MTJs se basent sur la magnétorésistance à effet tunnel – «Tunnel Magnetoresistance» (TMR), qui apparaît lorsque qu’une fine couche isolante est coincée entre deux matériaux ferromagnétiques (formant ainsi une structure de type MIM). Le courant passe par effet tunnel à travers la couche isolante. La résistance de celle-ci dépend de l’orientation relative de l’aimantation des deux couches ferromagnétiques. Des mé-

moires à base de MRAM ont déjà été réalisées avec succès, par exemple par Toshiba Corporation : 64 Mo en technologie 65 nm (Tsuchida et al., 2010). Ces dispositifs pourraient également être intéressants pour implémenter des neurones (Sharad et al., 2012). La génération actuelle de MRAM utilise le transfert de spin pour changer l'aimantation des couches ferromagnétiques (STT-MRAM). En éliminant le recours à un champ magnétique extérieur pour programmer les jonctions, celle-ci offre de nombreux avantages et permet notamment une plus grande densité d'intégration.

La structure d'un dispositif ferroélectrique est également de type MIM et le matériau ferroélectrique forme la couche isolante. Le courant passe par effet tunnel à travers celle-ci. Sa résistance électrique dépend de la polarisation ferroélectrique du matériau. La figure 2.29 montre une caractérisation synaptique du memristor ferroélectrique récemment publiée par l'Unité Mixte de Physique CNRS/Thales, qui semble prometteuse (Chanthbouala et al., 2012).

2.7 Discussion et perspectives

Ce chapitre a présenté les trois technologies qui ont directement été caractérisées et modélisées dans le cadre de ces travaux, à savoir le NOMFET, la PCM et la CBRAM. La technologie RRAM devrait faire l'objet des prochaines études en collaboration avec le CEA LETI, dans la logique de la continuité de l'évaluation des différentes technologies pour des applications synaptiques. La MRAM est quant à elle le cheval de bataille des équipes voisines du CNRS/Thales et de l'Institut d'Électronique Fondamentale (IEF) de l'Université Paris-Sud 11. Cette technologie est également activement développée par le CEA, via l'entreprise Crocus Technology, fondée en 2004. Toutes ces technologies présentent des caractéristiques intéressantes pour des applications synaptiques. Leur comparaison est un exercice difficile, car chaque technologie a des avantages et des inconvénients. Outre leurs caractéristiques synaptiques (nombre de niveaux et progressivité pour la potentialisation et la dépression), la complexité et la faisabilité de leur procédé de fabrication à l'échelle industrielle, ainsi que leur intégrabilité et densité d'intégration dans un circuit, sont autant d'éléments qui peuvent décider de l'avenir ou non d'une technologie. À noter que seules les technologies PCM et MRAM sont pour le moment considérées comme matures par l'International Technology Roadmap for Semiconductors (ITRS) (ITRS, 2011).

Le graphique de la figure 2.30 compare les dispositifs introduits précédemment sur six critères importants pour des synapses : la taille des cellules, le nombre de niveaux progressifs en potentialisation et dépression, l'énergie de programmation, la résistance R_{on} et le ratio $R_{\text{off}}/R_{\text{on}}$. Ces deux derniers paramètres doivent être aussi grands que possible pour limiter la consommation énergétique en lecture et surtout permettre une intégration dans une matrice de grande taille (Liang and Wong, 2010). Cette comparaison se base sur des données réelles de dispositifs expérimentaux publiés et non sur des extrapolations. Ainsi, il est remarquable de constater que la taille des cellules sur lesquelles les caractérisations multi-niveaux les plus complètes ont été faites, quelque soit la technologie, est supérieure à 200 nm. Si de la PCM à 2.5 nm a été démontrée (Liang et al., 2011) et que la limite théorique pour de la CBRAM est estimée à 2.5 nm (Zhirnov et al., 2011), seuls des comportements binaires ont été pour le moment observés à ces échelles là.

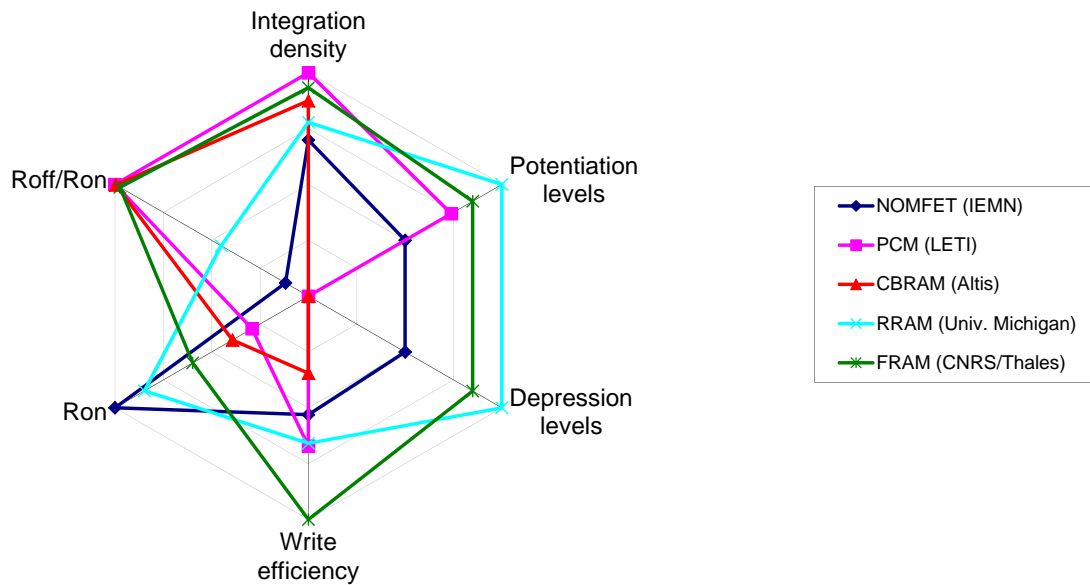


FIGURE 2.30 – Comparaison des différentes technologies pour une application synaptique : NOMFET (Alibart et al., 2010), PCM (Suri et al., 2011), CBRAM (Suri et al., 2012), RRAM (Jo et al., 2010) et FRAM (Chanthbouala et al., 2012). Les axes sont normalisés par rapport au meilleur dispositif et en échelle logarithmique.

Chapitre 3

Outils de modélisation et de simulation

Sommaire

3.1	Introduction	55
3.2	Simulateur évènementiel Xnet	56
3.2.1	Justification et objectifs	56
3.2.2	Principe de fonctionnement	56
3.2.3	Études de cas	61
3.2.4	Comparaison avec les simulateurs existants	64
3.3	Modélisation et simulation au niveau circuit	66
3.3.1	Présentation des langages de description matériel utilisés	67
3.3.2	Exemple de co-simulation Python/Cadence	68
3.4	Discussion et perspectives	70

3.1 Introduction

L'évaluation des technologies memristives pour des applications de traitement de données naturelles pose de nombreuses difficultés, à commencer par la modélisation des dispositifs. Pour une utilisation synaptique, un modèle doit tenir compte des aspects multi-niveaux, cumulatif et stochastique du dispositif. Réaliser un modèle physique rassemblant correctement tous ces aspects pour chaque technologie est compliqué. Les mécanismes physiques ne sont pas toujours bien compris et la multiplication des paramètres physiques rend leur ajustement sur des mesures expérimentales difficile. Alors qu'il existe des modèles très complets pour la mémoire à changement de phase – «Phase-Change Memory» (PCM) par exemple, un modèle semi-physique a été introduit dans le chapitre précédent principalement pour cette raison. Pour les autres technologies, un modèle comportemental, ou fonctionnel, permet d'émuler avec précision la ou les caractéristiques désirées, pour certaines conditions de programmation (par exemple amplitude et durée des impulsions de tension) déterminées à l'avance. Avec le «Nanoparticle-Organic Memory Field Effect Transistor» (NOMFET), nous avons montré que cette approche est efficace pour développer rapidement un modèle performant pour de l'exploration architecturale. Elle nécessite toutefois d'avoir accès aux dispositifs expérimentaux pour réaliser les caractérisations à inclure dans le modèle. La recherche des conditions de programmation optimales pour l'application envisagée se fait alors par cycles caractérisation-modélisation-simulation.

Démontrer la faisabilité d'un système de traitement de données naturelles basé sur ces modèles de dispositifs synaptiques suppose de pouvoir à la fois explorer les applica-

tions possibles et valider leur implémentation matérielle. La grande quantité de données (connaissance à priori) nécessaire pour réaliser de la détection ou de la reconnaissance d'objets suppose d'une part de pouvoir simuler efficacement des systèmes comprenant des millions de synapses pour être capable de mettre au point des applications intéressantes. D'autre part, une implémentation au niveau circuit est indispensable pour évaluer les performances réelles d'un tel système. Malgré cette contrainte, il est vite apparu que des simulations même fonctionnelles dans des langages de description matériel ne permettent pas de passer à l'échelle au niveau applicatif, en termes de performance et de productivité, ce qui m'a poussé à réaliser un outil de simulation de haut niveau pour l'exploration architecturale et applicative : le simulateur Xnet, présenté dans la première partie. En parallèle, la topologie en réseau de neurones et le codage de l'information impulsionnel ont motivé le développement d'une interface Python utilisant les outils de simulation de Cadence. Cette interface, présentée dans la deuxième partie, facilite la création au niveau circuit de réseaux complexes et la génération des signaux d'entrée adaptés.

3.2 Simulateur évènementiel Xnet

3.2.1 Justification et objectifs

Pour simuler des systèmes pour l'extraction de motifs audio ou vidéo, l'apprentissage, la reconnaissance et le suivi d'objets, les langages de description matériel tels que VHDL (IEEE, 2008) ou SystemC (IEEE, 2006) ne fournissent pas un niveau d'abstraction suffisant pour une exploration architecturale rapide et performante, qui implique généralement d'adapter la topologie du réseau, les paramètres neuronaux ou la règle d'apprentissage selon la tâche à effectuer. A l'opposé, les simulateurs de réseaux de neurones populaires dans la communauté des neurosciences tels que Neuron (Carnevale and Hines, 2006), Brian (Goodman and Brette, 2008) ou NEST (Gewaltig and Diesmann, 2007) peuvent procurer un haut niveau d'abstraction. Cependant, ils ne permettent pas d'intégrer facilement des contraintes liées au matériel ou manquent de flexibilité pour implémenter des fonctionnalités spécifiques pour le niveau applicatif.

3.2.2 Principe de fonctionnement

Xnet est conçu pour permettre la simulation d'architectures neuromorphiques impulsionnelles de grande échelle efficacement, avec un contrôle total de tous les détails d'implémentation des modèles de synapse et de neurone. Le simulateur est implémenté en C++11 (ISO, 2011), pour ses bonnes performances et ses capacités de programmation bas niveau. Le cœur de Xnet peut essentiellement être vu comme une combinaison entre un gestionnaire d'évènements et un jeu d'objets de base fournissant les fonctionnalités génériques pour la construction de modèles et de la topologie du réseau.

Traitement des évènements

Le moteur de simulation évènementiel de Xnet est décrit dans la figure 3.1 :

1. Les évènements sont triés par leur horodatage, avec le conteneur standard file de priorité de la bibliothèque standard C++ – «Standard Template Library» (STL). L'évènement en haut de la file est donc le prochain évènement programmé qui doit avoir lieu dans le temps virtuel du simulateur. Un évènement est défini par l'adresse du nœud d'origine qui l'a généré, l'adresse du nœud de destination et son horodatage virtuel. En plus, celui-ci contient un identifiant de type, utile lorsque plusieurs types d'évènements sont requis par les nœuds du réseau.

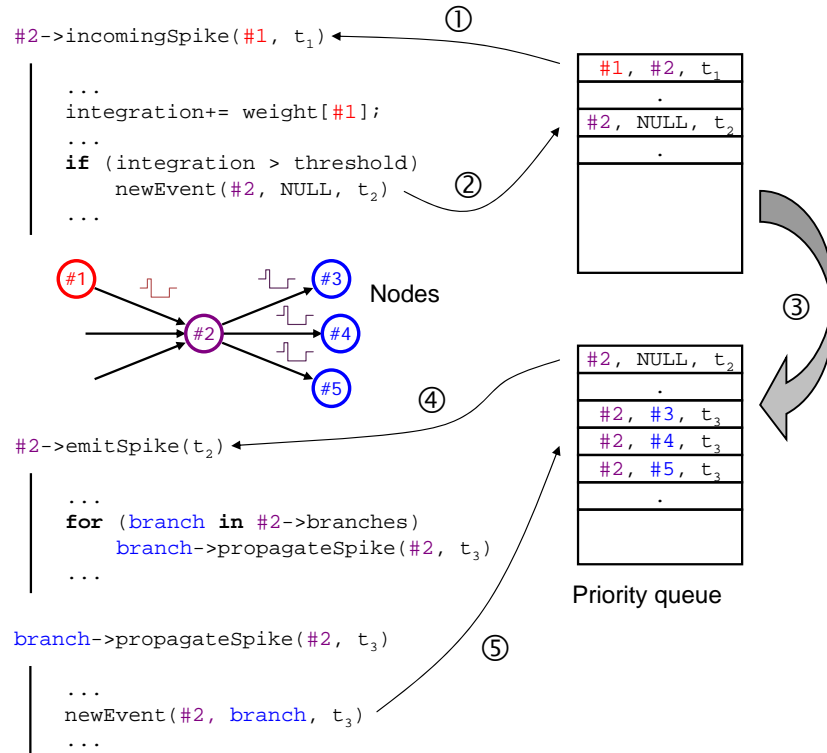


FIGURE 3.1 – Traitement des évènements dans Xnet. Le prochain évènement programmé est traité par son nœud de destination dans la routine `incomingSpike()` (1), qui peut générer des évènements internes au nœud (2) qui sont insérés dans la file de priorité (3). Les évènements internes sont traités dans la routine `emitSpike()` (4), qui peut émettre des évènements vers ses nœuds de sortie en appelant la routine `propagateSpike()` de ceux-ci (5).

Lorsque l'adresse de destination est connue, la routine `incomingSpike()` du nœud de destination est appelée.

- Le routine `incomingSpike()` traite l'évènement. Dans le cas d'un simple neurone «Leaky integrate-and-fire» (LIF), le poids synaptique de la connexion est ajouté à l'intégration du neurone (les connexions entrantes sont stockées dans un conteneur associatif standard de type `std::unordered_map<Node*, Synapse*>`, contenant les adresses des nœuds d'origine et l'adresse de leur objet `Synapse` associé, contenant l'état de la synapse). Si l'intégration du neurone atteint le seuil, un nouvel évènement est créé et inséré dans la file de priorité. Lorsqu'il ne contient pas d'adresse vers un nœud de destination (`NULL`), celui-ci est considéré comme un évènement interne au nœud courant. Les évènements en attente de traitement dans la file de priorité sont annulables. Cela est utile pour des modèles de neurone plus complexes où des évènements doivent être prédits et dont la prédiction peut changer en fonction d'évènements intermédiaires.
- Les évènements sont traités dans l'ordre jusqu'à ce que l'évènement précédemment inséré se retrouve en haut de la file.
- Si l'évènement n'a pas été annulé et comme il s'agit d'un évènement interne, la routine `emitSpike()` du nœud d'origine est appelée.
- La routine `emitSpike()` traite les évènements internes et crée les évènements vers les nœuds de sortie en appelant la routine `propagateSpike()`. `propagateSpike()` dans le nœud de sortie permet d'implémenter par exemple un délais de propagation pour la synapse, ou de modéliser des formes d'impulsions plus sophistiquées (voir la section 3.2.3).

Les nœuds du réseau héritent tous du même objet de base, `Node`, qui fournit une interface minimaliste pour le simulateur, incluant les routines `propagateSpike()`, `incomingSpike()` et `emitSpike()`. Les nœuds d'entrée du réseau sont de type `NodeEnv` et ceux-ci peuvent directement propager des impulsions vers leurs enfants en appelant `incomingSpike()`. Les modèles de neurone héritent normalement de l'objet de base `NodeNeuron` (lui-même hérité de `Node`), qui fournit des interfaces communes par exemple pour l'inhibition, la reconstruction des poids synaptiques, l'activité, etc... D'autres types de nœuds peuvent être créés, par exemple pour de la synchronisation, autorisant ainsi une grande flexibilité dans la description du réseau.

Modèle de base d'un neurone LIF

Dans cette section, l'explication théorique précédente est transposée à l'implémentation concrète d'un modèle de neurone LIF. Le listing 3.1 présente le code C++ minimaliste (et dépouillé de certaines lourdeurs syntaxiques) du neurone, avec inhibition latérale, période réfractaire et «Spike-Timing-Dependent Plasticity» (STDP). On y retrouve les 3 routines de base nécessaires pour le traitement des événements : `propagateSpike()`, `incomingSpike()` et `emitSpike()`. La fonction `propagateSpike()` implémente le délai synaptique `delay`, stocké dans l'objet `Synapse` et donc propre à chaque synapse. La fonction `incomingSpike()` met à jour l'instant d'activation de la synapse (temps pré-synaptique), utilisé par la règle d'apprentissage STDP. Elle met également à jour le potentiel de membrane du neurone (intégration) en y ôtant la valeur correspondante à la fuite depuis la dernière activation du neurone. Enfin, si celui-ci n'est pas en période réfractaire, le poids de la synapse est ajouté au potentiel de membrane. Si le potentiel de membrane est supérieur au seuil, un évènement est créé. Cet évènement est traité dans `emitSpike()`, qui remet l'intégration à zéro, calcule la période réfractaire, applique STDP aux synapses, inhibe les autres neurones et propage l'impulsion post-synaptique vers les neurones de sortie.

Le tableau 3.1 récapitule la liste des paramètres synaptiques et neuronaux constituant une simulation de base dans Xnet. La plupart des paramètres se rapportant aux synapses forment des couples (valeur moyenne, écart type) pour modéliser la variabilité inhérente aux dispositifs memristifs. Ceux-ci sont de la forme `Weight*`.

Listing 3.1 – Implémentation d'un modèle de neurone LIF avec inhibition latérale, période réfractaire et STDP. La fonction `stdp()` met à jour le poids de la synapse suivant la règle d'apprentissage.

```
1 void NodeNeuron::propagateSpike(Node* origin, Time_T timestamp) {
2     Synapse* synapse = mLinks[origin];
3     newEvent(origin, this, timestamp + synapse->delay);
4 }
5
6 void NodeNeuron::incomingSpike(Node* origin, Time_T timestamp) {
7     Synapse* synapse = mLinks[origin];
8     synapse->lastSpike = timestamp;
9
10    // Leak
11    mIntegration*= exp(-(timestamp - mLastSpikeTime)/mLeak);
12    mLastSpikeTime = timestamp;
13
14    if (timestamp >= mRefractoryEnd) {
15        // If not in refractory period:
16        // Integrate incoming spike
17        mIntegration+= synapse->weight;
18
19        if (mIntegration >= mThreshold)
```

```

20         // Fire!
21         newEvent(this, NULL, timestamp + mEmitDelay);
22     }
23 }
24
25 void NodeNeuron::emitSpike(Time_T timestamp) {
26     // Reset interation
27     mIntegration = 0.0;
28
29     // Enter refractory period
30     mRefractoryEnd = timestamp + mRefractory;
31
32     // STDP
33     for (auto it = mLinks.begin(), itEnd = mLinks.end(); it != itEnd; ++it) {
34         Synapse* synapse = (*it).second;
35         stdp(synapse, timestamp - synapse->lastSpike)
36     }
37
38     // Lateral inhibition
39     for_each(mLateralBranches.begin(), mLateralBranches.end(), bind(&lateralInhibition,
40         _1, timestamp));
41
42     // Propagate spikes to child nodes
43     for_each(mBranches.begin(), mBranches.end(), bind(&propagateSpike, _1, this,
44         timestamp));
45 }
46
47 void NodeNeuron::lateralInhibition(Time_T timestamp) {
48     if (mRefractoryEnd < timestamp + mInhibitRefractory)
49         // Add inhibition time to refractory period
50         mRefractoryEnd = timestamp + mInhibitRefractory;
51 }

```

Construction d'une topologie de réseau

Bien que les nœuds peuvent être instanciés et connectés individuellement pour former n'importe quelle topologie, la structure de base pour créer un groupe de neurones dans Xnet est l'objet `xcell`. Celle-ci implémente un groupe de neurones complètement connectés à leurs entrées avec une inhibition latérale globale. L'inhibition latérale peut être de n'importe quelle forme, la plus simple étant un «Winner-Take-All» (WTA) entre tous les neurones de la `xcell`. Comme Xnet est avant tout destiné à l'implémentation matérielle, l'architecture canonique d'une `xcell` est un «crossbar» de dispositifs synaptiques memristifs, comme illustré sur la figure 3.2.

Plusieurs `xcell` peuvent être groupées dans un objet `Layer` pour simplifier la conception de grands réseaux. Les entrées d'un `Layer` peuvent être distribuées entre ses `xcell` pour permettre l'apprentissage concurrent de sous-motifs spatialement localisés dans les stimuli d'entrée. Dans le cas de la modélisation d'applications de traitement en vision, les `xcell` dans un `Layer` peuvent être automatiquement distribués et connectés à différentes parties de l'image d'entrée pour former des macro-blocs, ou macro-pixels, qui peuvent par exemple apprendre des champs récepteurs locaux.

Les objets `xcell` et `Layer` permettent de faire des manipulations de groupe sur les neurones en une seule opération, comme la modification des paramètres neuronaux ou synaptiques, la sauvegarde ou le chargement de l'état des neurones et des synapses, etc...

TABLE 3.1 – Paramètres de base d’une simulation Xnet.

Paramètre du modèle	Type	Description
WeightsMinMean	Weight_T	Poids minimal moyen des synapses
WeightsMinStdDev	double	Écart type sur les poids minimaux
WeightsMaxMean	Weight_T	Poids maximal moyen des synapses
WeightsMaxStdDev	double	Écart type sur les poids maximaux
WeightsInitMean	Weight_T	Poids initial moyen des synapses
WeightsInitStdDev	double	Écart type sur les poids initiaux
WeightIncrementMean	Weight_T	Incrément moyen du poids lorsque la synapse subit une LTP
WeightIncrementStdDev	double	Écart type sur les incréments du poids
WeightIncrementDamping	double	Coefficient d’amortissement lors de l’incrément du poids (STDP multiplicative)
WeightDecrementMean	Weight_T	Décrément moyen du poids lorsque la synapse subit une LTD
WeightDecrementStdDev	double	Écart type sur les décréments du poids
WeightDecrementDamping	double	Coefficient d’amortissement lors du décrément du poids (STDP multiplicative)
IncomingDelayMean	Time_T	Temps de propagation moyen des synapses
IncomingDelayStdDev	Time_T	Écart type sur le temps de propagation
EmitDelayMean	Time_T	Délai moyen à l’émission du neurone
EmitDelayStdDev	Time_T	Écart type sur le délai moyen à l’émission
Threshold	double	Valeur initiale du seuil du neurone
Stdpltp	Time_T	Durée de la LTP
Leak	Time_T	Terme de fuite du neurone (constante de temps)
Refractory	Time_T	Période réfractaire du neurone après son déclenchement (intégration bloquée)
InhibitRefractory	Time_T	Période réfractaire du neurone après que celui-ci ait subi une inhibition

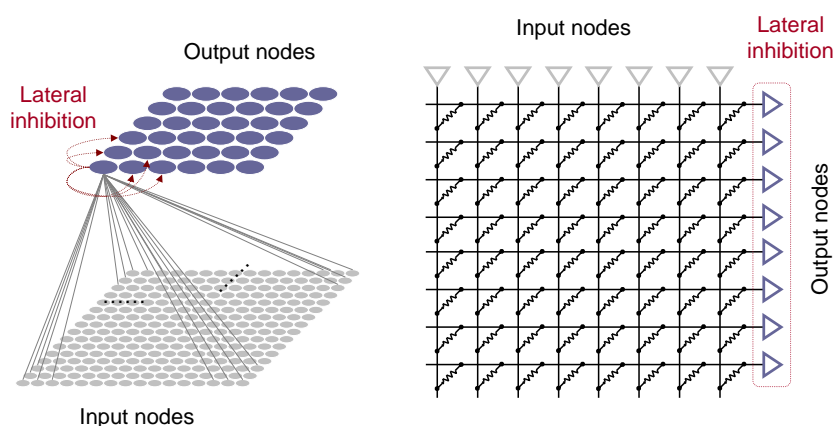


FIGURE 3.2 – Structure de base pour un groupe de neurones (*Xcell*), qui peut implémenter une inhibition latérale. Gauche : vue topologique. Droite : vue planaire du «crossbar» d’éléments synaptiques équivalent.

Génération des stimuli d’entrée

Les nœuds d’entrée du réseau (*NodeEnv*) peuvent gérer divers types de stimuli extérieurs via l’objet *Environment*, résumés sur la figure 3.3. Les images statiques et les stimuli

audio peuvent être codés temporellement pour émuler des capteurs de type rétine ou cochlée artificielle. Des données évènementielles, de type «Address-Event Representation» (AER), peuvent être chargées directement pour former les stimuli d'entrée.

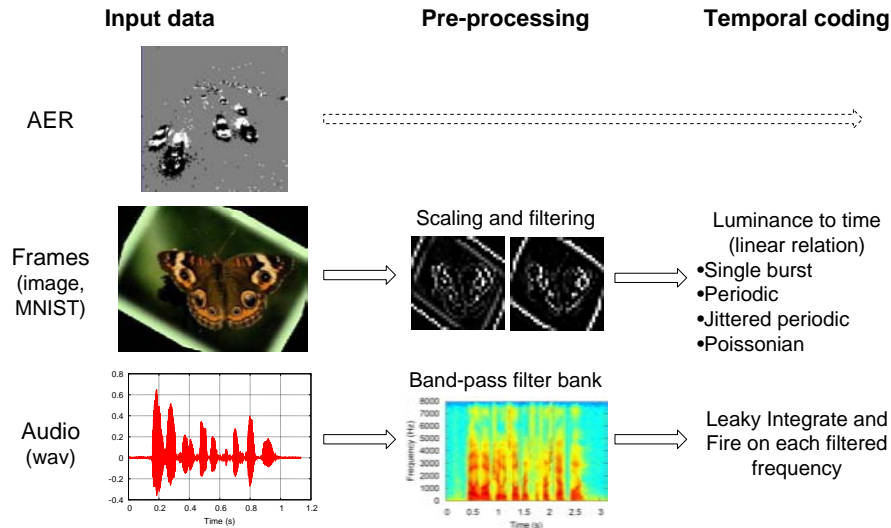


FIGURE 3.3 – Stimuli d'entrée possibles : données AER (`readAer()`), images statiques (`readFrame()`), signal audio (`readCochlea()`). Les étapes de pré-traitement et de codage temporel peuvent être adaptées pour émuler des capteurs matériels ou des unités de pré-traitement.

3.2.3 Études de cas

Dans cette partie, plusieurs exemples de simulations au niveau système effectuées avec Xnet sont présentées. Ils montrent la polyvalence du simulateur en intégrant diverses contraintes matérielles comme la durée des impulsions de tension, l'inhibition latérale, la modélisation système et comportementale de mémoire à changement de phase et l'implémentation d'opérateurs arithmétiques à l'aide d'un réseau de neurones impulsif.

Modélisation d'impulsions de durée non nulle

Bien que Xnet soit un simulateur évènementiel, simuler des impulsions de largeur non nulle est possible, sans la pénalité sur les performances qu'impliquerait l'introduction de pas de temps dans la simulation. La prise en compte de la largeur des impulsions est essentielle pour se rapprocher d'un niveau de modélisation électrique et faire le lien avec les simulations au niveau circuit. Pour se faire, les évènements dans Xnet possèdent un type et chaque nœud dans le réseau peut définir ses propres types. La figure 3.4 montre schématiquement comment un nœud simulant des impulsions de durée non nulle sur ses synapses peut être implémenté. Cela se fait avec l'introduction de deux types d'évènement, `EventOn` et `EventOff`, correspondant respectivement au front montant et au front descendant de l'impulsion. Au niveau des performances du simulateur, le doublement du nombre d'évènements pour la modélisation des impulsions a logiquement pour effet de multiplier par deux environ le temps de simulation.

Ce type de modèle est implémenté dans la classe `NodeNeuron_FirstOrder`, qui a été utilisée pour plusieurs publications (Querlioz et al., 2011a) (Querlioz et al., 2012a).

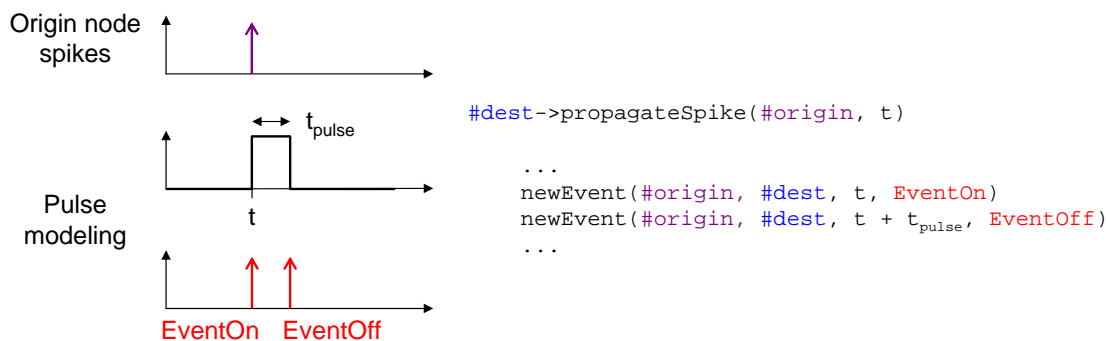


FIGURE 3.4 – Principe de la modélisation d’une impulsion de durée non nulle dans Xnet. Les évènements de type `EventOn` et `EventOff` modélisent respectivement le front montant et le front descendant de l’impulsion.

Extraction non supervisée de motifs

Dans cet exemple, nous simulons le développement de sélectivité pour un groupe de 10 neurones recevant des évènements d’une rétine 128×128 pixels, sensible aux variations relatives d’intensité lumineuse au niveau du pixel. Les neurones implémentent une forme spéciale de STDP, qui augmente le poids des synapses associées aux entrées activées juste avant l’impulsion post-synaptique et diminue le poids de toutes les autres synapses. Lorsque testé sur une séquence de 10 minutes de trafic sur une autoroute, le système développe des neurones sélectifs aux trajectoires des véhicules. Nous présenterons en détail cette simulation au chapitre «Étude d’architectures neuro-inspirées nouvelle génération pour le traitement de données naturelles», avec une étude du système complet constitué de deux couches de neurones et une analyse de la tolérance au bruit et à la variabilité.

Ce système peut être facilement simulé avec Xnet à l’aide du programme 3.2. D’abord, un `Environment` est instancié, avec deux filtres à l’échelle 1.0 ($= 128 \times 128$) qui seront automatiquement mis en correspondance avec les deux types d’évènement générés par la rétine (changements de luminosité positifs et négatifs (Lichtsteiner et al., 2008)). Une `Xcell` contenant 10 neurones avec des paramètres réglables est créée et connectée aux nœuds d’entrée de l’`Environment`. La séquence AER est ensuite lue par l’`Environment` et la simulation est démarrée (`run()`). Un `Monitor` permet d’enregistrer et de tracer l’activité des neurones, comme illustré sur la figure 3.5.

Listing 3.2 – Extraction de trajectoires de véhicules à partir d’un enregistrement AER.

```

1 Network net;
2 Environment env(net, 128, 128);
3 env.addMap(1.0, FilterAerPositive);
4 env.addMap(1.0, FilterAerNegative);
5
6 Xcell l1(net);
7 l1.populate<NodeNeuron_Behavioral>(10);
8 l1.setNeuronsParameter("Threshold", 1060000.0);
9 l1.setNeuronsParameter("Stdpltp", 15*TimeMs);
10 l1.setNeuronsParameter("Refractory", 517*TimeMs);
11 l1.setNeuronsParameter("InhibitRefractory", 10*TimeMs);
12 l1.setNeuronsParameter("Leak", 187*TimeMs);
13 l1.addInput(env, 0, 0, 0, 128, 128);
14
15 Monitor monit(net);
16 monit.add(l1);
17
18 for (unsigned int n = 0; n < 8; ++n) {
19     for (unsigned int i = 0; i < 85; ++i) {

```



```

20     env.readAer("events20051221T014416_freeway.mat.dat", 0, i*TimeS, (i+1)*TimeS);
21     net.run();
22     monit.update();
23 }
24
25 monit.logActivity("activity.dat", true);
26 l1.reconstructPatterns("res");
27 net.reset();
28 }

```

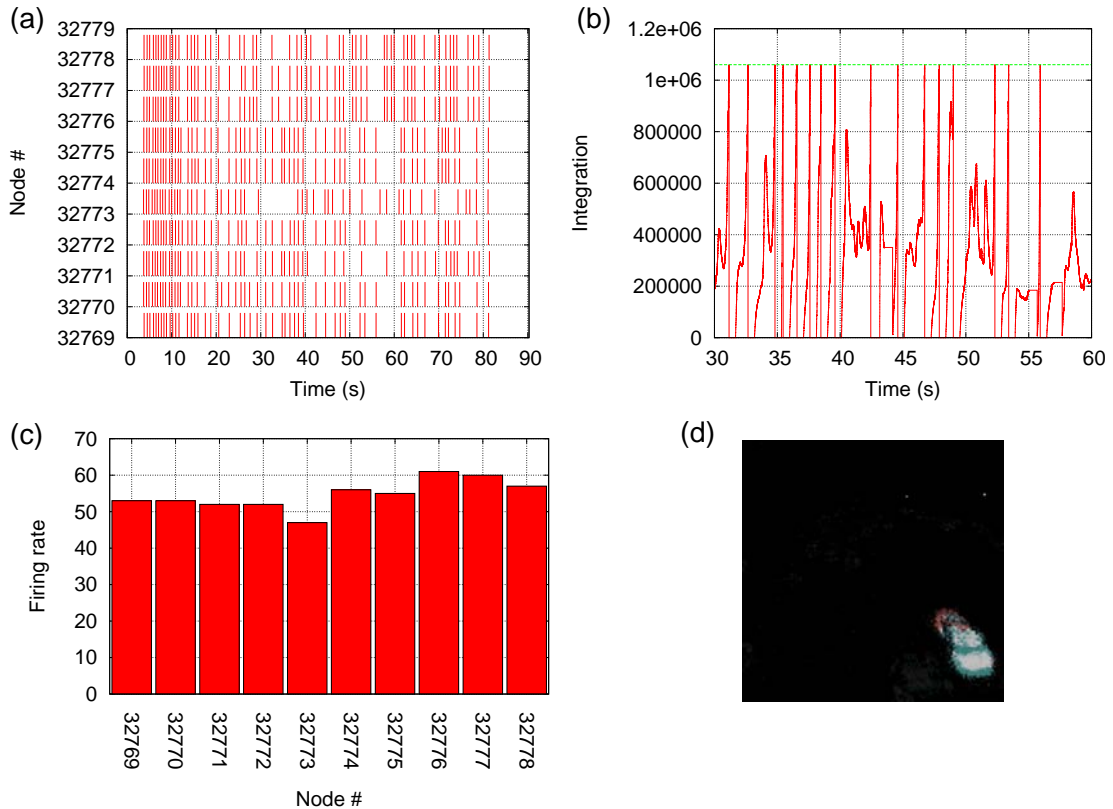


FIGURE 3.5 – Données en sortie d’une simulation typique. (a) Activité pour les 10 nœuds de sortie. (b) État interne de l’intégration d’un neurone. (c) Nombre d’activations pour les 10 nœuds de sortie. (d) Reconstruction des poids synaptiques pour un neurone de sortie.

Modélisation d’un système basé sur la PCM

L’étude de cas suivante présente l’implémentation de l’exemple précédent avec des dispositifs PCM pour les synapses. Le système complet est détaillé dans le chapitre «Implémentations matérielles sur des technologies memristives». Comme seule l’augmentation de la conductance des PCM est progressive avec des impulsions de tension identiques (voir figure 3.6(a)), un système utilisant deux dispositifs PCM par synapse a été inventé et simulé avec Xnet (figure 3.6(b)). Pour chaque connexion, l’état synaptique est stocké dans un objet spécifique `Synapse_PCM`, hérité de l’objet de base `Synapse`, qui contient deux jeux complets de paramètres pour le modèle comportemental de la figure 3.6(a).

Modélisation d’opérateurs logiques

Bien que cela sorte du cadre de ces travaux, Xnet a également été utilisé avec succès pour simuler des fonctions logiques implémentées avec des réseaux de neurones impul-

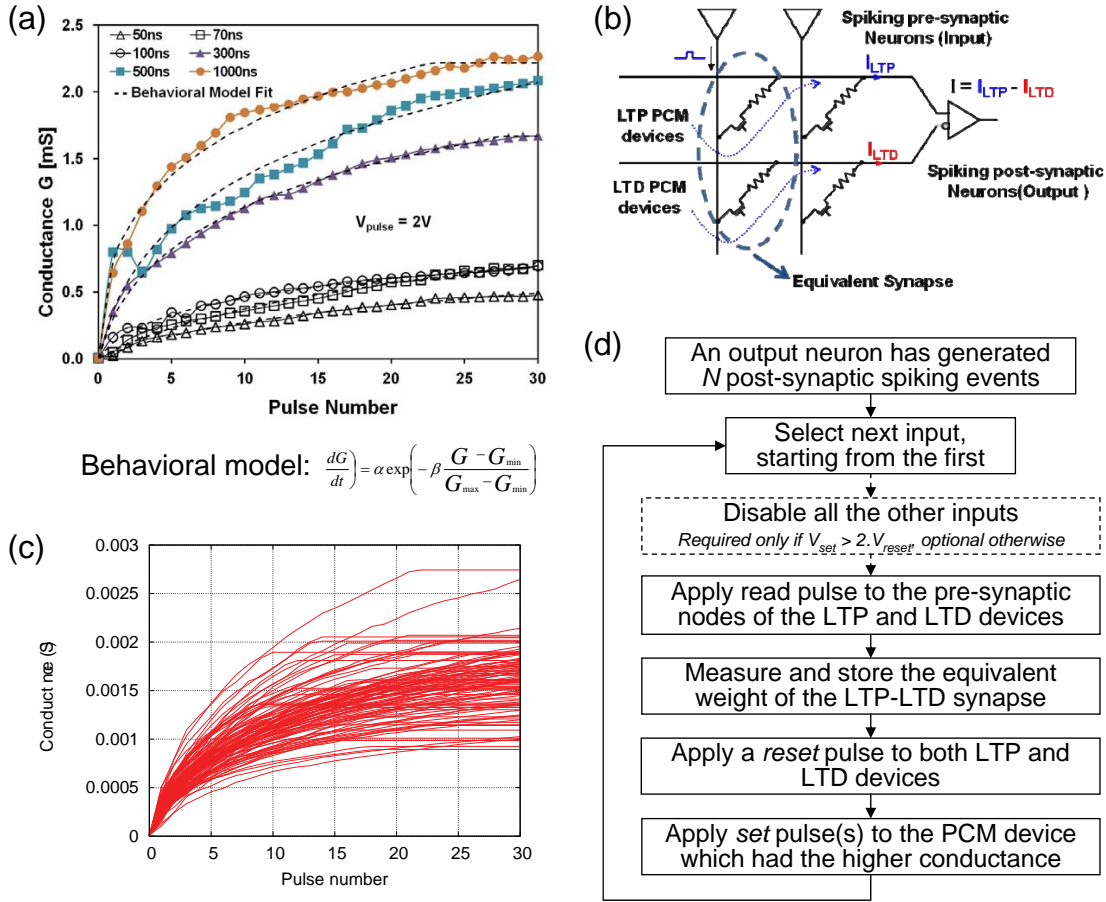


FIGURE 3.6 – Aperçu de la modélisation d’un système basé sur la PCM. (a) Interpolation de la caractéristique expérimentale des dispositifs (augmentation de la conductance du dispositif en fonction du nombre d’impulsions de tension appliquées). (b) Modèle de la synapse 2-PCM. (c) Variabilité simulée des dispositifs PCM sur 100 cycles de potentialisation. (d) Diagramme des opérations de rafraîchissement telles qu’implémentées dans Xnet.

sionnels. Il a notamment permis de simuler une fonction transformée de Fourier rapide – «Fast Fourier Transform» (FFT) sur des vecteurs d’entrée de taille 256, constituée d’un peu plus de 40 000 neurones (Gasnier, 2011). Le codage de l’information est fréquentiel. Ce travail a été effectué par des collaborateurs au Laboratoire d’électronique des technologies de l’information du Commissariat à l’énergie atomique et aux énergies alternatives (CEA) (CEA LETI).

3.2.4 Comparaison avec les simulateurs existants

Xnet ne se destine pas à simuler des systèmes biologiques, contrairement aux simulateurs pour les neurosciences, qui eux ne sont pas prévus pour simuler du matériel neuromorphique. Le simulateur Brian (Goodman and Brette, 2008) intègre toutefois de plus en plus de fonctionnalités tournées dans ce sens. Dans sa version 1.3 (sortie en février 2011), il est possible d’émuler approximativement l’application d’extraction de motifs, à partir d’un enregistrement AER, telle qu’implémentée dans Xnet. Cela demande d’utiliser des fonctionnalités expérimentales du simulateur. En outre, il n’est pas possible de remettre à zéro le potentiel de membrane des neurones via l’inhibition latérale, mais cette limitation est partiellement contournée en bloquant la contribution des impulsions pré-synaptiques à l’intégration des neurones durant leur période d’inhibition. Le potentiel de membrane des neurones revient alors à des valeurs proches de

zéro à la fin de l'inhibition grâce au terme de fuite. Le listing 3.3 contient le code Brian pour la simulation d'extraction de trajectoires de véhicules présenté dans la section précédente.

Listing 3.3 – Extraction de trajectoires de véhicules à partir d'un enregistrement AER avec Brian.

```

1 # Simulation parameters
2 parNbPass = 8
3 parNbNeuronsL1 = 60
4
5 # Synaptic parameters
6 WeightsMinMean = 0.001 * brian.volt
7 WeightsMaxMean = 1.0 * brian.volt
8 WeightsInitMean = 0.8 * brian.volt
9 WeightsInitStdDev = 0.16 * brian.volt
10 WeightIncrementMean = 0.1 * brian.volt
11 WeightDecrementMean = 0.05 * brian.volt
12
13 # Neural parameters
14 Threshold = 50.0 * brian.volt
15 StdpLtp = 15 * brian.ms
16 Refractory = 50 * brian.ms
17 InhibitRefractory = 10 * brian.ms
18 Leak = 200 * brian.ms
19
20 env = brian.experimental.neuromorphic.AERSpikeGeneratorGroup("/local/xnet_data/dvs/
    events20051221T014416_freeway.mat.dat", relative_time=False)
21
22 def lessOrEqual(inh, inhLim):
23     return 1.0*(inh <= inhLim)
24
25 inhLim = math.exp(-1.0) * brian.volt
26 eqs = """
27 dsyn/dt = -syn/1.0e-3 : volt
28 dinh/dt = -inh/InhibitRefractory : volt
29 dv/dt = (-v+syn*lessOrEqual(inh, inhLim))/Leak : volt
30 """
31 l1 = brian.NeuronGroup(N=parNbNeuronsL1, model=eqs, threshold=Threshold, reset=0.0 *
    brian.volt, refractory=Refractory)
32
33 winit = WeightsInitMean + WeightsInitStdDev*np.random.randn(len(env), len(l1))
34 synapses = brian.Connection(env, l1, 'syn', weight=winit)
35 inhibition = brian.Connection(l1, l1, 'inh', weight=1.0 * brian.volt)
36
37 eqsStdp = """
38 tPre : second
39 tPost : second
40 wUpdate : volt
41 """
42 stdp = brian.experimental.stdp.eventbased_stdp.EventBasedSTDP(synapses, eqs=eqsStdp,
    pre='tPre=t', post='tPost=t;w+=(WeightIncrementMean+WeightDecrementMean)*(tPre_
    <=tPost)*(tPre+_StdpLtp>=tPost)-WeightDecrementMean', wmin=WeightsMinMean,
    wmax=WeightsMaxMean)
43
44 monitorL1 = brian.SpikeMonitor(l1)
45 net = brian.Network(env, l1, synapses, inhibition, monitorL1)
46
47 for n in xrange(0, parNbPass):
48     for i in xrange(0, 85):
49         print "Learning from %d to %d (pass %d)" % (i, i+1, n+1),
50             nspikes = monitorL1.nspikes
51             net.run(1 * brian.second)

```

```

52     print "___[%d]" % (monitorL1.nspikes - nspikes)
53
54     net.reinit()
    
```

En termes de performances, les deux implémentations ont été comparées pour des simulations de réseaux de taille identique avec la même séquence en entrée. La configuration par défaut du simulateur Brian est utilisée (pour référence, la documentation indique que l'activation de l'ensemble des optimisations, ce qui inclut l'utilisation de code C compilé, peut résulter en un gain d'environ 30%). Les résultats du tableau 3.3 mettent bien en évidence les différences entre la simulation événementielle et la simulation par pas de temps. Avec un pas de temps de 0.1 ms, Brian est plus d'une fois et demi plus lent que Xnet pour simuler un réseau de 60 neurones et pas loin de 2 millions de synapses, tandis qu'il devient près de 3 fois plus rapide en multipliant le nombre de neurones et de synapses par 5. Le fait de multiplier par 5 le nombre de synapses multiplie par 5 le nombre d'évènements et la simulation prend logiquement 5 fois plus de temps avec Xnet. Ce résultat peut sembler finalement défavorable pour Xnet. Mais avec un pas de temps de 0.1 ms, la résolution temporelle de la simulation Brian est assez faible. Elle est tout juste suffisante par rapport aux constantes de temps du réseau, le pas de temps se rapprochant de 1% de la durée de la période de potentialisation à long terme – «Long-Term Potentiation» (LTP) de STDP. Une faible résolution peut en outre faire facilement apparaître des effets de bord tels que l'activation simultanée de plusieurs neurones malgré l'inhibition latérale, car elle ne permet pas de modéliser les délais aléatoires de propagation des synapses, ni les délais à l'émission des neurones. Ainsi, Xnet redevient plus de deux fois plus performant pour 300 neurones si la résolution temporelle de Brian est augmentée d'un ordre de grandeur, à 0.01 ms. La résolution temporelle des évènements dans Xnet est quant à elle fixée arbitrairement à 1 fs.

TABLE 3.3 – Performances comparées entre Xnet et Brian. Le nombre de synapses est de 1 966 080 pour 60 neurones et de 9 830 400 pour 300 neurones. Xnet traite en moyenne 2 millions d'évènements par seconde.

Simulation	Temps d'exécution	Conso. mémoire (max)
Xnet (60 neurones)	148 s ($\times 1.0$)	601 Mo ($\times 1.0$)
Brian (60 neurones, $\Delta T = 0.1$ ms)	249 s ($\times 1.68$)	967 Mo ($\times 1.61$)
Brian (300 neurones, $\Delta T = 0.1$ ms)	312 s ($\times 2.11$)	2425 Mo ($\times 4.03$)
Xnet (300 neurones)	855 s ($\times 5.78$)	2743 Mo ($\times 4.56$)
Brian (300 neurones, $\Delta T = 0.01$ ms)	2000 s ($\times 13.5$)	2472 Mo ($\times 4.11$)

3.3 Modélisation et simulation au niveau circuit

Nous proposons dans cette partie une méthodologie pour faire de la simulation de circuit et de l'analyse de résultats dans un contexte neuronal, ce pour quoi les outils de conception assistée par ordinateur (CAO) pour les circuits intégrés sont inadaptés. Pour ce faire, le langage de programmation Python est utilisé pour préparer les simulations et les analyser.

Le type d'architecture que l'on cherche à concevoir et à valider au niveau circuit est en effet radicalement différent des architectures de traitement classiques. Les méthodes de test numériques usuelles, consistant à vérifier les données générées en sortie de l'algorithme implémenté - ou du programme exécuté - sur le circuit, ne permettent pas facilement de vérifier les résultats issus d'un apprentissage, qui sont variables et non déterministes du fait du traitement analogique de l'information sur des dispositifs

aux paramètres fortement variables et intrinsèquement stochastiques. La génération des vecteurs de test pour de l'apprentissage est également compliquée par la quantité de données à traiter et leur codage temporel.

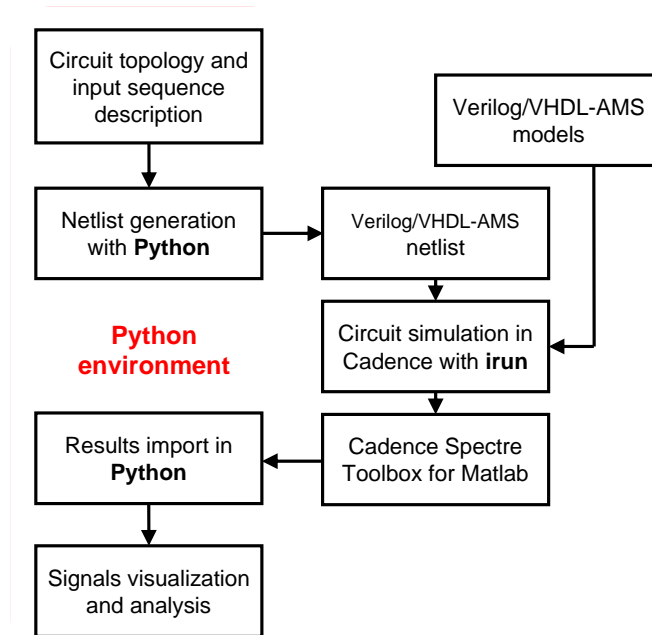


FIGURE 3.7 – Flot pour la simulation hybride nano-dispositif memristif/CMOS avec Python et Cadence.

Pour pallier à ces difficultés et faciliter la conception d’une architecture hybride nano-dispositif memristif/«Complementary Metal Oxide Semiconductor» (CMOS), un flot de simulation développé en Python utilisant les outils de simulation de Cadence a été développé (représenté sur la figure 3.7). Celui-ci reprend le concept introduit dans la section 2.2.4 pour le NOMFET et remplace Matlab par Python, qui a l’avantage d’offrir un modèle de programmation objet et de disposer de meilleures interfaces pour la manipulation de données non numériques (fichiers texte par exemple). L’interface entre Python et Cadence est rendu possible par la commande `irun`. Cet utilitaire, fourni par Cadence, permet de lancer des simulations multi-langage (Verilog, VHDL, SystemC...) et multi-environnement (analogique, numérique, mixte) en une seule commande, qui peut prendre en entrée l’ensemble des fichiers sources et options de configuration pour la simulation.

3.3.1 Présentation des langages de description matériel utilisés

Pour réaliser des simulations de circuits mixtes analogique/numérique, les deux langages de description matériel universels sont Verilog-AMS et VHDL-AMS. Ils disposent globalement des mêmes fonctionnalités et du même support sur les principaux simulateurs mixte analogique/numérique – «Analog and Mixed-Signal» (AMS). Il est possible dans une même simulation d’instancier des modèles dans les deux langages de manière transparente. Contrairement à Verilog, VHDL sépare la déclaration et l’implémentation d’une entité, ce qui facilite beaucoup la description et la simulation d’une architecture sur plusieurs niveaux de granularité. Cela permet de concevoir l’architecture en partant de sa description fonctionnelle de haut niveau jusqu’à son implémentation au niveau transistor. Cette méthodologie est illustrée sur la figure 3.8. VHDL permet également pour chaque entité de créer des configurations, spécifiant quelles architectures utiliser

pour les entités plus bas dans la hiérarchie. Ainsi, la sélection des branches à simuler sur la figure 3.8 ne nécessite aucune intervention dans le code de la description matérielle.

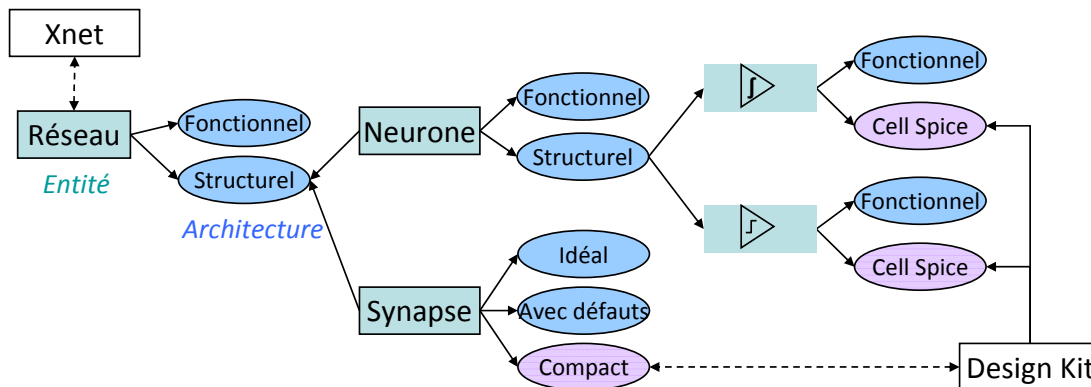


FIGURE 3.8 – Méthodologie de conception d'une architecture en partant d'un modèle à haut niveau d'abstraction pour arriver au niveau transistor par raffinement successif, avec le langage VHDL-AMS.

3.3.2 Exemple de co-simulation Python/Cadence

Dans cette partie, nous nous proposons de simuler le mécanisme d'apprentissage STDP sur des dispositifs PCM, en implémentant au niveau circuit le mécanisme de programmation qui est décrit dans le chapitre «Implémentations matérielles sur des technologies memristives». Le modèle semi-physique de la PCM introduit avec la présentation de la technologie est utilisé.

Pour référence, l'implémentation se décompose en 8 fichiers VHDL :

- `model_pcm.vhms` : modèle semi-physique d'un dispositif PCM ;
- `neuron_ctrl.vhd` : machine d'état globale contrôlant l'inhibition des neurones ;
- `neuron_in_ctrl.vhd` : machine d'état d'un neurone d'entrée ;
- `neuron_in.vhms` : interface analogique d'un neurone d'entrée ;
- `neuron_out_ctrl.vhd` : machine d'état d'un neurone de sortie ;
- `neuron_out.vhms` : interface analogique d'un neurone de sortie ;
- `xbar_pcm.vhms` : matrice de synapses PCM, interfacée avec les neurones d'entrée et de sortie ;
- `cell.vhms` : système complet.

Le listing 3.4 montre comment créer une «netlist» Verilog-AMS avec la classe Python `VamsNetlist`. La figure 3.9 correspond au schéma équivalent après instantiation des deux entités U0 et U1. La classe `stimuli` permet quant à elle de générer des signaux logiques en VHDL. En passant par Python pour générer la «netlist» de haut niveau («top-level»), la création des signaux d'entrée à partir de données extérieures, comme une base d'images ou de vidéos, est facilitée (VHDL et Verilog disposent tous deux de fonctions basiques pour la gestion de fichiers, mais bien trop limitées pour faire du pré-traitement sur des images par exemple). L'utilisation d'un script simplifie également l'automatisation de simulations, pour faire varier des paramètres ou tester un grand nombre de vecteurs d'entrée. Enfin, il offre un gain de productivité par rapport à une saisie sur schéma via une interface graphique, pour l'édition des paramètres des instances ou le changement de la taille des bus d'entrée/sortie notamment.

Listing 3.4 – Génération d'une «netlist» avec un script Python. La création des stimuli d'entrée, contenu dans la variable `stimu`, n'est pas détaillée (voir figure 3.10 pour leur représentation graphique).

```

1 # Simulation parameters
2 SizeIn = 4 # Number of input neurons (= size of input stimuli)
3 SizeOut = 1 # Number of output neurons
4 Tclk = 1e-5 # Clock period
5 Tstimu = 1e-3 # Duration of a stimuli
6 StimuliLength = 60 # Number of stimuli presented during learning
7 UpdateLimit = 20 # Number of weight updates on a 2-PCM synapse before a refresh
8
9 # Generate netlist
10 netlist = VamsNetlist.VamsNetlist()
11 netlist.addElectrical("rd_voltage", "wr_voltage", "rst_voltage", "thres_voltage")
12 netlist.addBus("in_event", SizeIn)
13 netlist.addDcSource("Vrd", ["rd_voltage", "gnd"], 0.1)
14 netlist.addDcSource("Vwr", ["wr_voltage", "gnd"], 1.0)
15 netlist.addDcSource("Vrst", ["rst_voltage", "gnd"], 8.0)
16 netlist.addDcSource("Vthres", ["thres_voltage", "gnd"], 3.5e-5)
17 netlist.addInstance("U0", {
18     "clk": "clk", "rst": "rst", "in_event": "in_event", "out_event": "out_event", "
19     rd_voltage": "rd_voltage", "wr_voltage": "wr_voltage", "rst_voltage": "
20     rst_voltage", "thres_voltage": "thres_voltage", "ref": "gnd"
21 }, "cell", {
22     "SizeIn": SizeIn, "SizeOut": SizeOut, "Tread": netlist.time(100e-9), "Twrite":
23     netlist.time(300e-9), "Treset": netlist.time(100e-9), "Tleak": netlist.time
24     (0.35*Tstimu), "Tltp": 0.35*Tstimu/Tclk, "Tinhbit": 0.35*Tstimu/Tclk, "
25     UpdateLimit": UpdateLimit
26 })
27 netlist.addInstance("U1", {"clk": "clk", "rst": "rst", "in_event": "in_event"}, "
28     ctrl_stimu")
29
30 # Generate control stimuli
31 stm = Stimuli.Stimuli("ctrl_stimu")
32 stm.addClock("clk", Tclk)
33 stm.addLogical("rst", [0, 1e-5])
34 stm.addLogical("in_event", stimu)

```

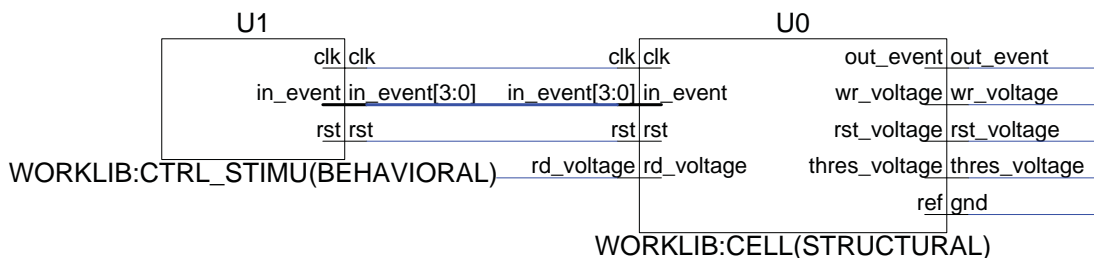


FIGURE 3.9 – Schéma de la «netlist» correspondante au listing 3.4, obtenue avec l’outil SimVision de Cadence.

La configuration de la simulation se fait avec la classe `AmsSimulation`, listing 3.5.

Listing 3.5 – Configuration de la simulation et exécution.

```

1 # Generate simulation
2 ams = AmsSimulation.AmsSimulation("/local/simus/cell")
3 ams.attachNetlist(netlist)
4 ams.includeStimuli(stm)
5 ams.include("../model_pcm.vhms")
6 ...
7 ams.spectre.analysis['tran'].addOption("stop", (StimuliLength + 1)*Tstimu)
8 ams.addTclProbe()
9 ams.simulate()

```

Enfin, le listing 3.6 montre comment il est possible de récupérer les signaux de la simulation et de les afficher directement avec Python grâce au module Pylab. Les deux figures générées par ce script sont montrées figure 3.10 et 3.11.

Listing 3.6 – Récupération des résultats.

```

1 # Display simulation results
2 pylab.figure(1)
3 ams.plot("in_event[3:0]", append=True)
4 ams.plot("U0:inhibit", append=True)
5 ams.plot("out_event", append=True)
6
7 pylab.figure(2)
8 pylab.subplot(SizeIn + 1, 1, 1)
9 ams.plot("out_event")
10
11 for i in xrange(0, SizeIn):
12     pylab.subplot(SizeIn + 1, 1, i + 2)
13     pylab.hold(True)
14     pylab.plot(Rgst_p[i][0].time, 1.0/numpy.array(Rgst_p[i][0].V), 'r', Rgst_n[i][0].
15               time, 1.0/numpy.array(Rgst_n[i][0].V), 'b')
16     pylab.ylabel("in_%" + str(i))

```

La figure 3.10 montre de haut en bas les stimuli pour les quatre neurones d'entrée (pré-synaptiques), le signal d'inhibition et le signal de sortie du neurone post-synaptique. Trois entrées sur 4 sont activées aléatoirement toutes les millisecondes, avec une probabilité d'activation de la dernière entrée inférieure de 50% aux 3 autres. Les synapses ayant activé le neurone sont renforcées avec la règle d'apprentissage STDP, tandis que celles qui n'ont pas été activées sont affaiblies. Sur la figure 3.11, on vérifie bien que la dernière synapse voit son poids équivalent diminuer plus rapidement que les 3 autres, puisqu'elle subit plus souvent une dépression à long terme – «Long-Term Depression» (LTD), ce qui a pour effet d'augmenter la conductance du dispositif LTD. Passé une trentaine de présentations de triplets (après 30 ms), seule l'activation rapprochée des trois premières entrées déclenche l'activation du neurone, qui est devenu sélectif au motif le plus probable. Entre 35 et 40 ms, le mécanisme de rafraîchissement des synapses reprogramme les poids synaptiques en minimisant la conductance des dispositifs PCM. Cette procédure, déjà simulée dans Xnet, peut être étudiée ici avec un modèle fonctionnel du circuit. Cela constitue un premier pas vers sa validation qui nécessiterait la conception des blocs analogiques au niveau transistor.

3.4 Discussion et perspectives

Le simulateur Xnet a depuis sa création mi-2010 été utilisé pour les résultats de 9 publications scientifiques (Bichler et al., 2012a,b; Suri et al., 2012; Querlioz et al., 2012b,a) (Suri et al., 2011, 2012) (Bichler et al., 2011; Querlioz et al., 2011a). Il a également été utilisé dans le cadre d'un stage sur l'implémentation d'une FFT avec des opérateurs neuromorphiques (Gasnier, 2011) et est actuellement utilisé pour un autre stage à l'Institut d'Électronique Fondamentale (IEF) sur le développement d'architectures à base de jonctions tunnel magnétiques – «Magnetic Tunnel Junctions» (MTJs). Xnet dispose aujourd'hui d'une assise solide et de nombreuses fonctionnalités pour faciliter la conception et l'analyse de réseaux de neurones impulsifs. Il supporte bien la comparaison avec les simulateurs existants tout en étant le seul à notre connaissance à réellement se focaliser à la fois sur l'implémentation et l'aspect applicatif pour des architectures à base de dispositifs memristifs.

La plateforme de co-simulation Python/Cadence et la méthodologie sous-jacente ont été développées pour faciliter la conception et la validation d'un système de traitement

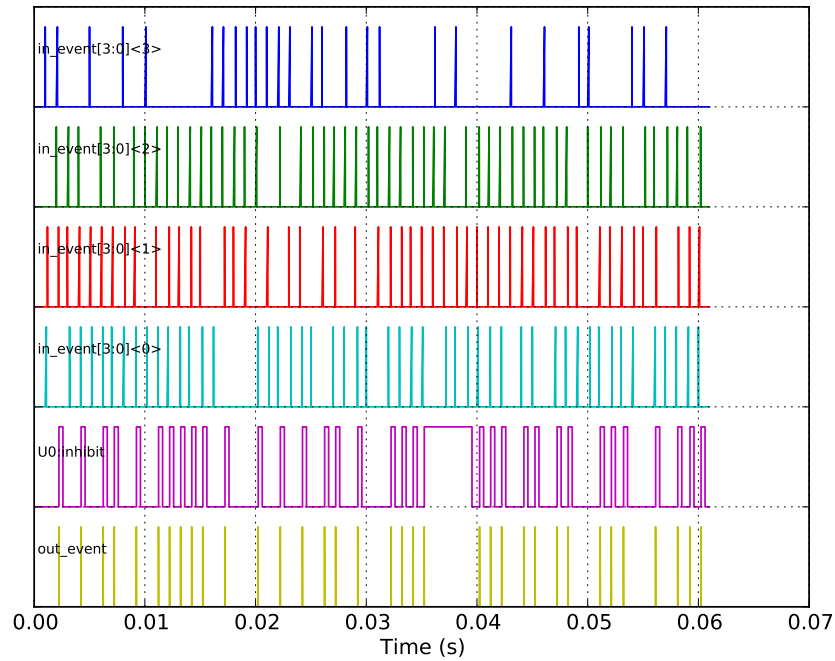


FIGURE 3.10 – Visualisation des signaux d'entrée et de la sortie du neurone pour le système d'apprentissage à base de PCM.

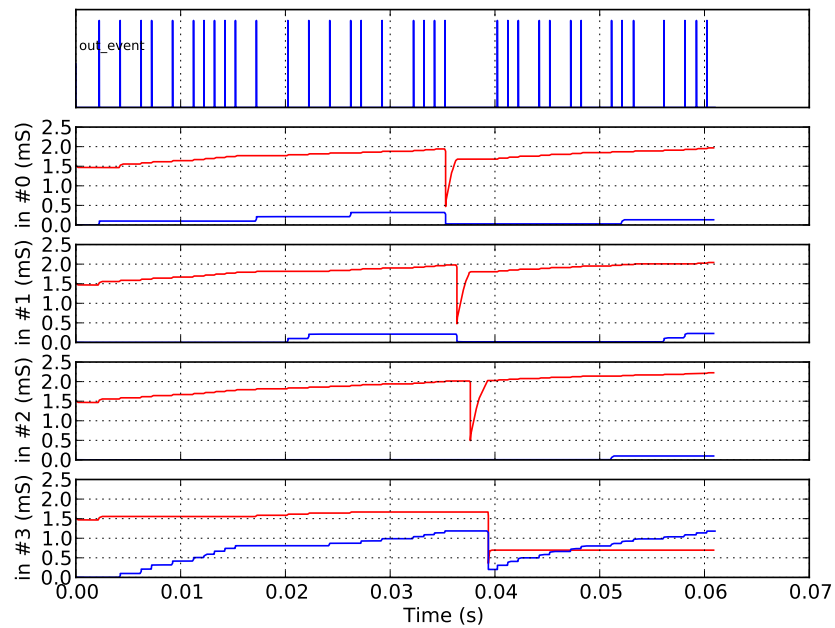


FIGURE 3.11 – Évolution de la conductance des dispositifs PCM.

de données naturelles complet au niveau circuit, qui aurait été dans un premier temps mis au point avec Xnet.

A l'heure actuelle, dans le monde, aucun système hybride à base de dispositifs memristifs de taille conséquente n'a été réalisé avec succès (ou rendu public), malgré un effort considérable dans le projet SyNAPSE, financé par la Defense Advanced Research Projects Agency (DARPA). La difficulté est en grande partie technologique, mais aussi méthodologique. La tendance générale des implémentations proposées de STDP est d'émuler les mesures biologiques dans l'idée de s'affranchir de l'aspect applicatif, qui est considéré comme acquis de par les capacités cognitives du cerveau biologique. Les approches explorées, en particulier dans le projet de la DARPA, sont

donc de type «bottom-up». La méthodologie que nous proposons est au contraire de type «top-down», puisqu'elle part d'un ensemble d'applications possibles pour aller par raffinements successifs à une implémentation circuit adaptée à celles-ci. Nous espérons que cette méthodologie ainsi que les outils associés pourront être mis en œuvre avec succès pour la réalisation future d'un prototype à moyenne échelle (plusieurs milliers de synapses).

Chapitre 4

Étude d'architectures neuro-inspirées nouvelle génération pour le traitement de données naturelles

Sommaire

4.1	Introduction	73
4.2	Extraction non supervisée de motifs visuels	74
4.2.1	Méthodologie	75
4.2.2	Principe de l'apprentissage	77
4.2.3	Apprentissage de motifs à partir de capteurs de vision dynamiques	81
4.2.4	Robustesse et immunité au bruit	89
4.3	Détection de motifs auditifs dans du bruit	90
4.3.1	Méthodologie	91
4.3.2	Résultats des simulations	94
4.4	Apprentissage et classification de motifs	97
4.4.1	Méthodologie	97
4.4.2	Performances comparées avec un Perceptron	102
4.5	Discussion et perspectives	104

4.1 Introduction

Ce chapitre présente en détail trois applications de traitement de données naturelles étudiées dans ces travaux : l'apprentissage et l'extraction de motifs visuels, de motifs auditifs et la classification. Nous développons et mettons en œuvre les idées présentées dans l'introduction et le premier chapitre. Nous montrons comment - et avec quel codage temporel de l'information - réaliser de l'apprentissage non supervisé de motif en utilisant la règle d'apprentissage «Spike-Timing-Dependent Plasticity» (STDP). Ce chapitre répond au premier enjeu majeur évoqué dans l'introduction : montrer que des applications concrètes sont ainsi réalisables pour le traitement des données naturelles. Il répond également en partie au troisième enjeu, qui est de démontrer la tolérance de telles architectures aux variabilités synaptiques. Ce dernier point est essentiel pour envisager des implémentations sur des dispositifs memristifs, qui à l'échelle nanométrique sont intrinsèquement fortement variables.

Les applications présentées dans ce chapitre se veulent être relativement indépendantes vis-à-vis du nano-dispositif memristif utilisé pour implémenter les synapses. Ainsi, nous préférons dans la suite parler simplement de synapse, plutôt que de désigner explicitement telle ou telle technologie memristive. L'implémentation concrète des architectures présentées dans ce chapitre sur des nano-dispositifs fait l'objet du chapitre suivant, «Implémentations matérielles sur des technologies memristives». Dans ce chapitre, nous réintroduisons et réutilisons le modèle déjà utilisé pour les caractéristiques de potentialisation et de dépression des dispositifs mémoire à changement de phase – «Phase-Change Memory» (PCM) et mémoire résistive – «Resistive RAM» (RRAM) (section 2.3.3), comme modèle générique d'évolution du poids synaptique.

4.2 Extraction non supervisée de motifs visuels

La grande majorité des senseurs de vision et des systèmes de traitement associés fonctionnent à partir d'une capture d'images statiques, chaque image passant dans l'ensemble de la chaîne de traitement. Cependant, pour de nombreuses applications, notamment liées au mouvement, les images successives contiennent une quantité importante d'information redondante, qui doit tout de même être traitée. Cela a un coût élevé en termes de puissance de calcul, de temps de traitement et de consommation électrique. Pour de l'analyse de mouvements, les changements locaux au niveau des pixels et leur instant d'apparition sont les seules informations vraiment nécessaires, ce qui peut ne représenter qu'une faible fraction de l'ensemble des données transmises par un senseur de vision conventionnel. L'élimination de cette information redondante est d'ailleurs à la base des algorithmes de compression vidéo. Le fait que chaque pixel d'une image ait le même temps d'exposition contraint également la plage dynamique et la fréquence d'échantillonnage du capteur.

Les rétines impulsionnelles silicium, qui s'inspirent du fonctionnement des rétines biologiques, sont une réponse directe à cette problématique. Au lieu d'envoyer des images, une rétine artificielle utilise l'«Address-Event Representation» (AER) pour transmettre les impulsions en réponse aux changements locaux temporels et/ou spatiaux (Lichtsteiner et al., 2008; Zaghoul and Boahen, 2004). Dans ces capteurs, également appelés senseurs de vision dynamique AER, les adresses des impulsions sont transmises de manière asynchrone (en temps réel) sur un bus série. Bien que relativement récentes, plusieurs types de rétines impulsionnelles ont déjà été réalisés, avec des résolutions de l'ordre de 128×128 pixels.

Les avantages indéniables de ces rétines dynamiques du point de vue de la quantité d'information à traiter les rendent également plus difficile à appréhender, car la plupart des algorithmes de traitement de la vision classiques ne sont pas exploitables ou sont inefficaces (Delbruck, 2008). Les convolutions classiques par exemple sont difficiles à implémenter, car l'activation des pixels est asynchrone et le flot de données AER est continu. Des réseaux convolutionnels basés sur des impulsions ou de l'AER existent (Pérez-Carrasco et al., 2010). Les poids des noyaux de convolution sont souvent appris hors ligne en utilisant une architecture classique à base d'images statiques. Plus fondamentalement, ces approches se basent sur la fréquence moyenne de chaque pixel, ignorant ainsi toute l'information contenue dans les instants d'arrivée relatifs entre les impulsions (Guyonneau et al., 2004).

Pour surpasser ces difficultés, nous proposons une approche qui exploite pleinement la nature impulsionnelle et asynchrone d'une rétine et capable d'extraire des motifs visuels complexes, se recouvrant temporellement, de manière robuste et non supervisée. Nous montrons une nouvelle manière de traiter des stimuli impulsionnels, issus de scènes réelles enregistrées avec un capteur AER, avec un algorithme de type STDP.

Des séquences de mouvements d'objets individuels peuvent être apprises à partir de séquences complexes avec un réseau de neurones impulsionnels «feedforward» multicouches non supervisé. Ces travaux qui s'appuient sur les concepts introduits dans le premier chapitre (Masquelier et al., 2008), exploitent les temps d'arrivée relatifs des impulsions et montrent des performances excellentes, en considérant la simplicité et la nature non supervisée du mécanisme d'apprentissage mis en œuvre. Ces résultats seront exploités dans le chapitre suivant pour des implémentations matérielles efficaces avec des dispositifs memristifs.

Ces travaux ont été publiés dans (Bichler et al., 2011) Bichler et al. (2012a).

4.2.1 Méthodologie

Un réseau de neurones impulsionnels faisant de la reconnaissance de motif à partir de données issues d'une rétine AER est simulé. Pour cela, nous utilisons le simulateur évènementiel C++ Xnet pour l'ensemble des simulations. La simulation évènementielle est particulièrement bien adaptée au traitement d'un flux de données AER, contrairement aux simulateurs de réseaux de neurones classiques basés sur des pas de temps, qui sont généralement plus focalisés sur le réalisme de la modélisation biologique que sur une simulation matérielle efficace. Notre simulateur est capable de traiter les données d'une rétine AER 128×128 pixels en quasi temps réel sur un «Central Processing Unit» (CPU) standard de bureau.

Règle d'apprentissage

La règle d'apprentissage, commune pour l'ensemble des simulations présentées dans cette section, est une règle STDP simplifiée. Toutes les synapses d'un neurone sont affaiblies de la même ampleur lorsqu'elles reçoivent une impulsion post-synaptique. Ceci à l'exception des synapses qui ont été activées avec une impulsion pré-synaptique un court instant avant. Celles-ci sont au contraire fortement renforcées, comme illustré sur la figure 4.3. L'affaiblissement des autres synapses est systématique, qu'elles soient activées ensuite ou non et même si elles n'ont jamais été activées avant. Ce comportement ne peut donc pas être entièrement modélisé avec la fonction «fenêtre» STDP classique $\Delta w = f(t_{post} - t_{pre})$.

Nous verrons dans la suite que cette règle, dite «simplifiée», a en fait des conséquences importantes pour l'apprentissage. Cela est discuté de manière approfondie dans les sections 4.2.2 et 4.2.3.

Dans le cas d'un renforcement du poids synaptique, ou potentialisation à long terme – «Long-Term Potentiation» (LTP), la forme générale de l'équation de mise à jour du poids est la suivante :

$$\Delta w_+ = \alpha_+ \cdot \exp\left(-\beta_+ \cdot \frac{w - w_{min}}{w_{max} - w_{min}}\right) \quad (4.1)$$

Dans le cas d'un affaiblissement, ou dépression à long terme – «Long-Term Depression» (LTD), l'équation est similaire :

$$\Delta w_- = \alpha_- \cdot \exp\left(-\beta_- \cdot \frac{w_{max} - w}{w_{max} - w_{min}}\right) \quad (4.2)$$

où $\alpha_+ > 0$, $\beta_+ \geq 0$, $\alpha_- < 0$ et $\beta_- \geq 0$ sont quatre paramètres. w est le poids de la synapse et peut évoluer entre w_{min} (> 0) et w_{max} . En fonction des deux paramètres β , on peut modéliser une règle de mise à jour des poids purement additive ($\beta = 0$) ou pseudo-multiplicative, ce qui permet de modéliser différents types d'implémentations matérielles (ou logicielles) (Querlioz et al., 2011b) sans altérer le principe de fonctionnement du processus d'apprentissage proposé. Nous avons montré des exemples de

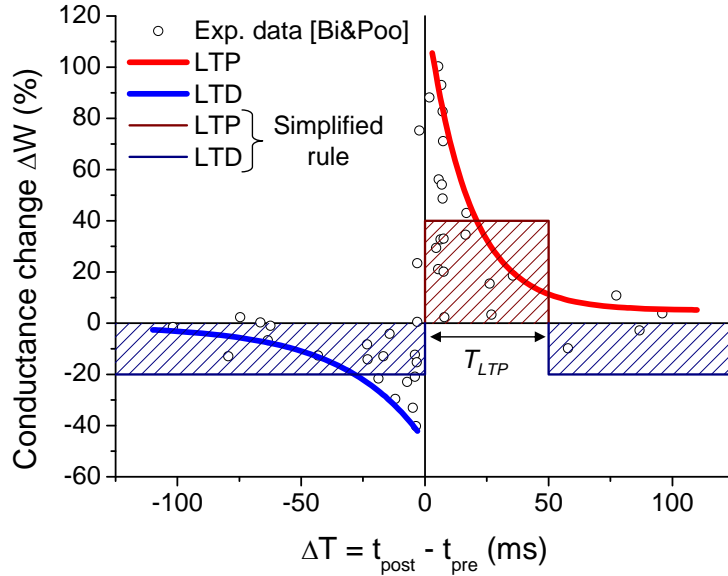


FIGURE 4.1 – Règle d'apprentissage STDP : les synapses subissent une LTP lorsque $0 \leq t_{post} - t_{pre} \leq T_{LTP}$ et une LTD sinon.

dispositifs memristifs au chapitre «Technologies mémoires émergentes et memristivité» pouvant être modélisés avec ce modèle (PCM et RRAM notamment).

Modèle de neurone impulsif

Dans notre simulateur évènementiel, un évènement impulsif au temps t_{spike} est modélisé par une impulsion de Dirac $\delta(t - t_{spike})$. Entre deux impulsions, l'intégration u du neurone de type «Leaky integrate-and-fire» (LIF) est la solution de l'équation différentielle suivante :

$$u + \tau_{leak} \cdot \frac{du}{dt} = 0 \quad (4.3)$$

L'état de l'intégration du neurone n'a besoin d'être mis à jour que lors de l'évènement impulsif suivant, au temps t_{spike} , où le poids synaptique w de l'impulsion entrante est ajouté à l'intégration :

$$u = u \cdot \exp\left(-\frac{t_{spike} - t_{last_spike}}{\tau_{leak}}\right) + w \quad (4.4)$$

Lorsque l'intégration u atteint le seuil du neurone, un nouvel évènement impulsif est créé et envoyé à l'ensemble des neurones de sortie. L'intégration est alors réinitialisée à zéro et ne peut plus augmenter avant la fin de la période réfractaire T_{refrac} .

Inhibition latérale

Lorsqu'un neurone se déclenche, celui-ci désactive tous les autres neurones durant une période $T_{inhibit}$, durant laquelle aucune impulsion entrante n'est intégrée. Cette période d'inhibition s'ajoute à la période réfractaire des neurones récemment activés, dans le cas où $T_{inhibit} < T_{refrac}$. Parce que les neurones fuient, si $T_{inhibit} \gg T_{leak}$, ceux-ci sont complètement réinitialisés après la période d'inhibition latérale.

Données AER

Ces simulations utilisent les données AER enregistrées avec le capteur TMPDIFF128 DVS (Lichtsteiner et al., 2008) et téléchargées à partir de (DVS128 Dynamic Vision Sensor Silicon Retina data, 2011) ou générées dans le même format. Un ensemble de données AER consiste en une liste d'événements, avec pour chaque événement l'adresse du pixel correspondant de la rétine, son temps d'émission et son type. Pour le capteur TMPDIFF128, un pixel génère un événement dès que l'intensité lumineuse qu'il reçoit atteint un seuil, positif ou négatif. Ainsi, selon le signe du changement d'intensité lumineuse, les événements peuvent être de type ON ou de type OFF, correspondant respectivement à une augmentation ou une diminution de luminosité.

4.2.2 Principe de l'apprentissage

Dans cette section, nous présentons tout d'abord un exemple d'apprentissage simple de trajectoires de balles à partir d'une séquence AER générée par ordinateur. Nous passerons à la démonstration d'un apprentissage sur une séquence réelle enregistrée à partir d'une rétine AER 128×128 pixels dans la section suivante.

Extraction de trajectoires partielles

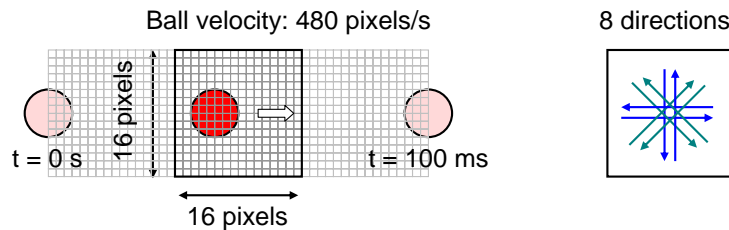


FIGURE 4.2 – Caractéristiques des trajectoires de balles générées par ordinateur, utilisées comme stimuli d'entrée du réseau. Une balle se déplace dans l'une des 8 directions à la vitesse de 480 pixels/s sur une grille de 16×16 pixels. Les événements AER sont générés en imitant les propriétés d'une rétine impulsionnelle artificielle.

Pour la première expérience, nous avons généré par ordinateur 8 séquences AER, chacune représentant la trajectoire d'une balle dans une direction différente, comme illustré sur la figure 4.2. Les caractéristiques des données générées sont identiques à celles enregistrées avec le capteur TMPDIFF128, avec une résolution plus faible de 16×16 pixels. Chaque pixel d'entrée nécessite deux synapses, pour transmettre les événements de type ON et OFF, ce qui donne un nombre total de $2 \times 16 \times 16 = 512$ adresses d'entrée. Notre réseau de neurones est constitué de 48 neurones de sortie, avec 512 synapses par neurone (voir figure 4.3), chaque synapse étant identifiée par son événement correspondant. Une inhibition latérale est également implémentée : chaque neurone inhibe l'ensemble des autres neurones durant une période $T_{inhibit}$ lorsqu'il est activé.

Lorsqu'un neurone se déclenche à l'instant t_{spike} , la STDP augmente le poids des synapses les plus récemment activées, entre $t_{spike} - T_{LTP}$ et t_{spike} , et diminue le poids de l'ensemble de ses autres synapses. Cela a pour effet d'augmenter la sensibilité du neurone au motif spécifique qui a mené à son activation. Il est ainsi plus susceptible de se déclencher pour un motif similaire, corrélé, dans le futur. Comme le neurone comprend un terme de fuites, seule la contribution des événements ayant activé une majorité de synapses fortement renforcées sur une période courte a des chances d'élever l'intégration du neurone au-dessus du seuil. Cela assure que le neurone est uniquement sensible à un motif spécifique, typiquement un groupe d'événements fortement corrélés

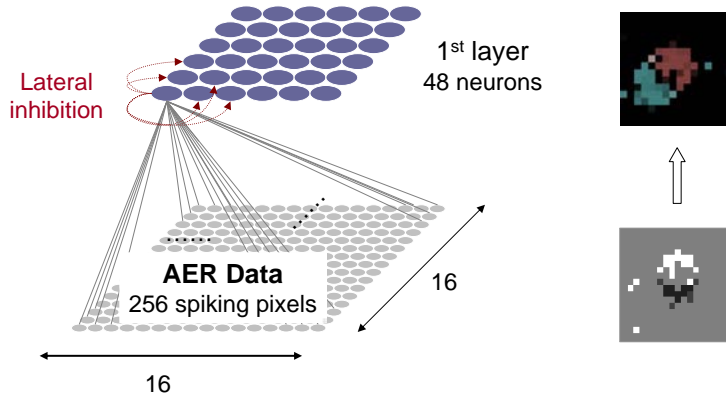


FIGURE 4.3 – Topologie du réseau de neurones pour l'extraction de trajectoires partielles. Il s'agit d'un réseau «feedforward» avec une couche complètement connecté et une inhibition latérale complète, de chaque neurone à tous les autres neurones. Il n'y a donc pas d'inhibition spatiale spécifique entre les neurones.

temporellement. Les figures 4.4 et 4.5 montrent l'activité du réseau avant et après apprentissage, respectivement.

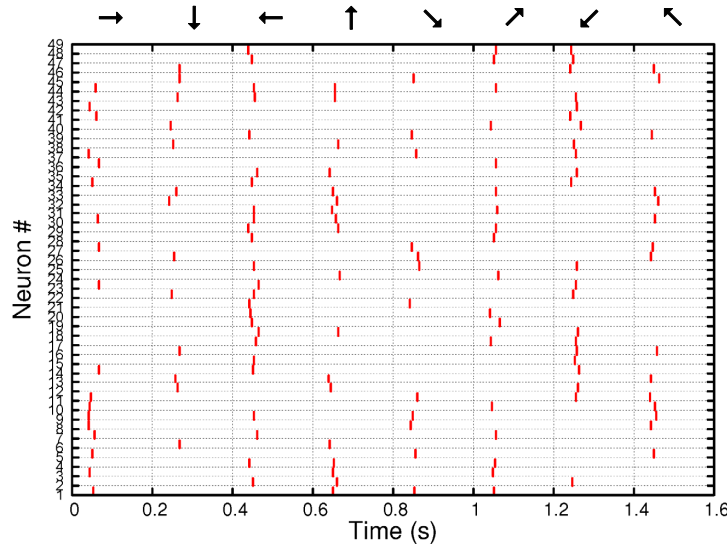


FIGURE 4.4 – Les évènements impulsionnels émis par les neurones de sortie (axe vertical) en fonction du temps (axe horizontal). La direction des trajectoires de balles présentées est indiquée en haut de la figure. Initialement, le poids des synapses est égal en moyenne à 80% du poids maximal. Les neurones sont donc très actifs et ne présentent pas de sélectivité pour les différentes trajectoires, comme cela peut être vu lorsque les 8 stimuli AER sont présentés dans l'ordre, un toute les 200 ms.

Deux mécanismes assurent un apprentissage compétitif et complémentaire des neurones (Masquelier et al., 2009). Le premier est l'inhibition latérale, qui est fondamentale pour permettre à plusieurs neurones d'apprendre plusieurs motifs. Sans inhibition latérale, tous les neurones finiraient par apprendre le même motif. Le temps d'inhibition $T_{inhibit}$ contrôle en fait l'intervalle de temps minimum entre les parties dans lesquelles une trajectoire peut être décomposée, chaque partie étant apprise par un neurone différent, comme vu sur la figure 4.6.

Le deuxième mécanisme est la période réfractaire du neurone lui-même, qui permet avec l'inhibition latérale d'adapter la dynamique de l'apprentissage (et spécifiquement le nombre de parties utilisées pour décomposer une trajectoire) à la dynamique des stimuli

d'entrée (la vitesse du mouvement). Si par exemple le mouvement est lent par rapport à la période d'inhibition, la période réfractaire des neurones assure qu'un neurone unique ne peut pas remonter une trajectoire entière en se déclenchant de manière répétitive et en adaptant ses poids aux stimuli d'entrée évoluant lentement. Un tel neurone serait «glouton», dans le sens où il enverrait continuellement des trains d'impulsions en réponse à diverses trajectoires, ne laissant aucune chance aux autres neurones de se déclencher et d'apprendre quelque chose d'utile.

Après l'apprentissage, on peut désactiver l'inhibition latérale pour vérifier que les neurones sont suffisamment sélectifs pour ne se déclencher qu'avec le motif qu'ils ont appris, comme cela peut se déduire de la reconstruction des poids. A partir de ce moment, même en poursuivant la présentation de stimuli avec STDP activé, l'état de la plupart des neurones reste stable sans inhibition latérale. Quelques uns s'adaptent vers d'autres motifs, ce qui est attendu puisque STDP est toujours en action. Mais plus important, aucun neurone «glouton» n'apparaît, qui se déclencherait pour de multiples motifs sans sélectivité.

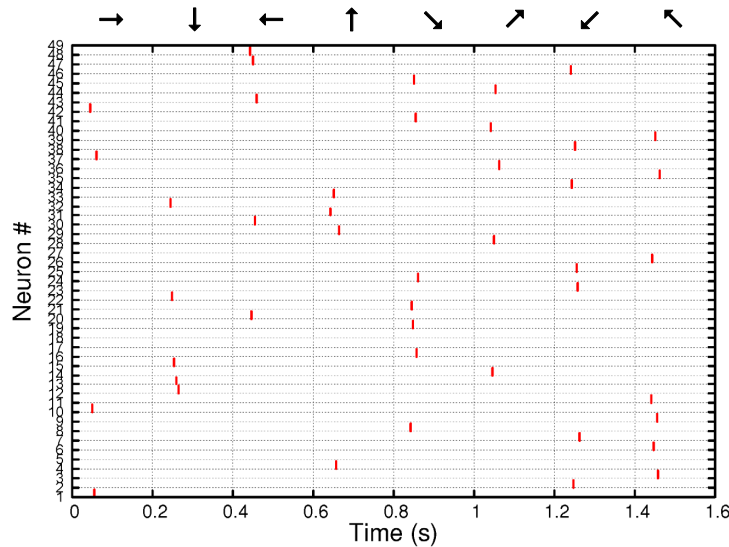


FIGURE 4.5 – Après 2000 présentations dans un ordre aléatoire, les 8 stimuli AER sont à nouveau présentés, un toutes les 200 ms. Maintenant, chaque neurone se déclenche uniquement pour un morceau de trajectoire particulier.

Les paramètres neuronaux de cette simulation sont résumés dans le tableau 4.1. En général, les paramètres pour les poids synaptiques ne sont pas critiques pour le mécanisme d'apprentissage proposé (voir tableau 4.2). Seules deux conditions importantes doivent être vérifiées :

1. Dans toutes nos simulations, Δw_+ devait rester supérieur à Δw_- . Dans les premières phases de l'apprentissage, la LTD est initialement plus forte que la LTP. Les neurones ne sont pas sélectifs et tous leurs poids synaptiques sont en moyenne diminués. Toutefois, parce que les poids initiaux sont distribués aléatoirement et grâce à l'inhibition latérale, les neurones deviennent forcément à partir d'un certain moment plus sensibles à certains motifs qu'à d'autres. La LTP induite par le motif préféré doit alors surpasser la LTD induite par les autres motifs, ce qui n'est pas forcément garanti si Δw_+ est trop faible. D'un autre côté, si Δw_+ est trop grand, les poids synaptiques ne sont plus diminués en moyenne et les synapses des neurones ne peuvent pas être suffisamment affaiblies pour permettre à ceux-ci de devenir sélectifs.
2. On doit avoir $\Delta w < (w_{max} - w_{min})$. Toutefois, une haute précision n'est pas

nécessaire. 4 à 5 bits par poids sont suffisants, puisque dans nos simulations $\Delta w_+ = 2.\Delta w_- = (w_{max} - w_{min})/10$.

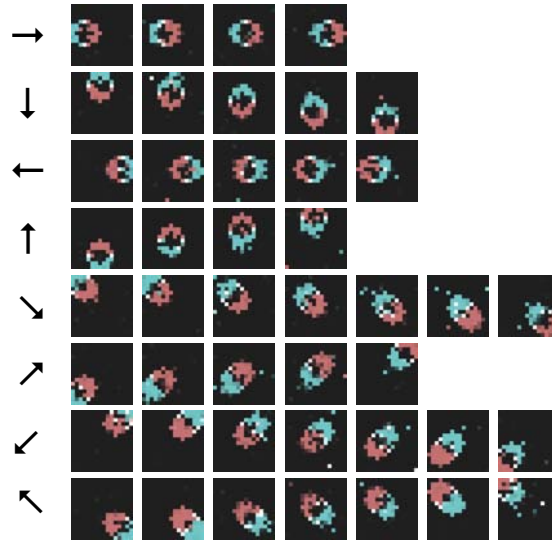


FIGURE 4.6 – Reconstruction des poids après apprentissage. Les reconstructions sont ordonnées du premier au dernier neurone activé pour chaque trajectoire, d'après l'activité reportée sur la figure 4.5. Le rouge représente les synapses renforcées connectées aux sorties positives (ON) des pixels et le bleu représente les synapses renforcées connectées aux sorties négatives (OFF) des pixels. Lorsque les synapses ON et OFF sont toutes les deux renforcées pour un pixel donné, la couleur résultante est gris clair.

TABLE 4.1 – Description et valeur des paramètres des neurones pour l'extraction de trajectoires partielles.

Paramètre	Valeur	Effet
I_{thres}	40000	Le seuil affecte directement la sélectivité des neurones. La valeur maximale du seuil est limitée par T_{LTP} et τ_{leak} .
T_{LTP}	2 ms	La taille temporelle du morceau à apprendre par un neurone.
T_{refrac}	10 ms	Doit être supérieur à $T_{inhibit}$, mais inférieur à la période typique de répétition du motif appris par le neurone.
$T_{inhibit}$	1.5 ms	Intervalle de temps minimum entre les morceaux dans lesquels une trajectoire peut être décomposée.
τ_{leak}	5 ms	La constante de fuite doit être légèrement supérieure à la durée typique des motifs à apprendre.

Il est remarquable qu'en moyenne, il y a 1.4 fois plus de neurones activés pour les trajectoires diagonales que pour les trajectoires horizontales et verticales. Ce chiffre est consistant avec le rapport de distance entre ces deux types de trajectoires, qui vaut $\sqrt{2}$.

Évolution génétique

Trouver des valeurs optimales pour les paramètres neuronaux I_{thres} , T_{LTP} , T_{refrac} , $T_{inhibit}$ et τ_{leak} peut être une tâche difficile. Toutefois, puisque tous les neurones dans une même couche partagent les mêmes paramètres, cela ne fait que 5 paramètres différents par couche qu'il faut adapter à un type particulier de stimuli. Cette tâche peut être accomplie efficacement avec un algorithme génétique, à condition que l'activité cible du réseau puisse être correctement formulée. Pour ce faire, des instances multiples du réseau de neurones avec des paramètres «mutés» aléatoirement apprennent en parallèle

TABLE 4.2 – Moyenne et écart type pour les paramètres synaptiques, pour toutes les simulations de cette section. Les paramètres sont choisis aléatoirement pour chaque synapse au début des simulations, en utilisant une distribution normale.

Paramètre	Moyenne	Écart type	Description
w_{min}	1	0.2	Poids minimum (normalisé).
w_{max}	1000	200	Poids maximum.
w_{init}	800	160	Poids initial.
α_+	100	20	Incrément du poids.
α_-	50	10	Décrément du poids.
β_+	0	0	Saturation à l'incrément.
β_-	0	0	Saturation au décréement.

et un score pour chaque instance est calculé à la fin de l'apprentissage. Les paramètres des instances avec les meilleurs scores sont alors «mutés» à nouveau pour la génération suivante. Le score est calculé en comparant l'activité de la deuxième couche avec l'activité de référence obtenue avec un étiquetage à la main de la séquence. Les trains d'impulsions d'activité sont convolués avec la fonction gaussienne $\exp(-(t/\tau)^2)$ pour former un signal continu. La valeur absolue de la différence des signaux résultants, pour l'activité de sortie du réseau et l'activité de référence, est intégrée et normalisée. Des paramètres corrects peuvent être obtenus en moins de 10 générations, avec 80 instances et 8 gagnants par génération, même pour l'apprentissage de motifs complexes dans de grands réseaux comme celui présenté dans la section suivante. Dans toute cette partie, les paramètres neuronaux sont obtenus avec une évolution génétique.

4.2.3 Apprentissage de motifs à partir de capteurs de vision dynamiques

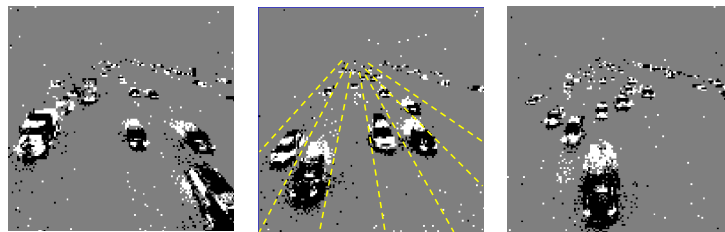


FIGURE 4.7 – Illustration du jeu de données utilisé pour l'apprentissage de motifs complexes : véhicules passant sous un pont sur l'autoroute 210 à Pasadena. Les pixels blancs représentent les événements ON et les pixels noirs les événements OFF. Au centre de l'image, les délimitations des voies de circulation sont matérialisées avec des lignes en pointillés. Cette séquence AER, ainsi que plusieurs autres, sont disponibles sur ([DVS128 Dynamic Vision Sensor Silicon Retina data, 2011](#)).

Dans cette section, nous montrons comment le mécanisme d'apprentissage introduit précédemment peut être utilisé pour extraire des trajectoires plus complexes, se recouvrant dans le temps, directement à partir d'une rétine AER. Le stimulus utilisé dans cette partie a été enregistré à partir du capteur TMPDIFF128 DVS par le groupe de Tobi Delbruck et est disponible sur ([DVS128 Dynamic Vision Sensor Silicon Retina data, 2011](#)). Il représente des véhicules passant sous un pont sur l'autoroute 210 à Pasadena. La séquence dure 78.5 s, contient 5.2M évènements avec une fréquence d'évènement moyenne de 66.1k évènements par seconde. La figure 4.7 montre un rendu de la séquence obtenu avec le logiciel jAER ([jAER Open Source Project, 2011](#)), qui accumule les évènements pendant une courte durée pour former une image. Compter le

nombre de véhicules passant sur chaque voie de circulation en regardant la séquence à l'œil nu est une tâche pratiquement impossible, car il n'y a pas de repère autre que le mouvement des véhicules pour distinguer les voies et le trafic est relativement dense.

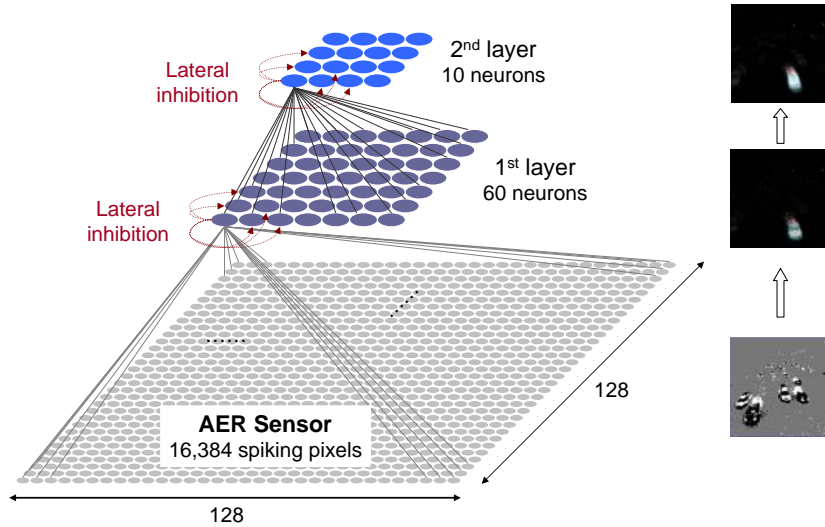


FIGURE 4.8 – Topologie du réseau de neurones pour l'apprentissage de trajectoires complexes, directement à partir de données enregistrées avec le capteur AER. Il s'agit d'un réseau de type «feedforward», de deux couches, complètement connecté, avec de l'inhibition latérale, de chaque neurone vers tous les autres neurones. Il n'y a pas d'inhibition spatialement localisée entre les neurones. La couche du bas représente le capteur AER et n'est pas considérée comme une couche du réseau de neurones.

Le réseau de neurones utilisé pour cette simulation est décrit à la figure 4.8. Il s'agit d'un réseau de type «feedforward», de deux couches, complètement connecté, avec 60 neurones sur la première couche et 10 neurones sur la seconde. Le nombre total de synapses dans ce système est de $2 \times 128 \times 128 \times 60 + 60 \times 10 = 1\,966\,680$. Chacune des couches peut être implémentée avec un ou plusieurs «crossbar» de dispositifs memristifs.

Deux stratégies d'apprentissage ont été successivement testées dans la suite, toutes les deux complètement non supervisées. Dans la première, qui peut être appelée apprentissage «global», les deux couches apprennent simultanément, l'inhibition latérale étant toujours activée. Dans la deuxième stratégie, seule la première couche est active dans une première étape. Une fois que les trajectoires apprises sont stables, l'inhibition latérale est supprimée et STDP est désactivée pour cette couche. La deuxième couche n'est autorisée à apprendre qu'après cette étape et l'inhibition latérale est également supprimée après apprentissage. Dans cette stratégie, il n'y a plus d'inhibition latérale du tout dans le réseau après que chaque neurone se soit spécialisé et nous montrerons les avantages de cette méthode pour l'extraction exhaustive de trajectoires se recouvrant temporellement.

Apprentissage global

Dans la première stratégie d'apprentissage, les deux couches apprennent en même temps et l'inhibition latérale est toujours active. Si l'on ne considère que la première couche, cette expérience est identique à la précédente avec les trajectoires des balles. Il est remarquable que, malgré le fait que les trajectoires se recouvrent continuellement dans le temps, le trafic étant plutôt dense, le mécanisme d'apprentissage décrit précédemment arrive toujours à extraire les morceaux de trajectoires associés avec une seule voie de circulation, comme le confirme la reconstruction des poids de la première couche sur la figure 4.9. Parce qu'il n'y a pas de corrélation particulière entre les temps

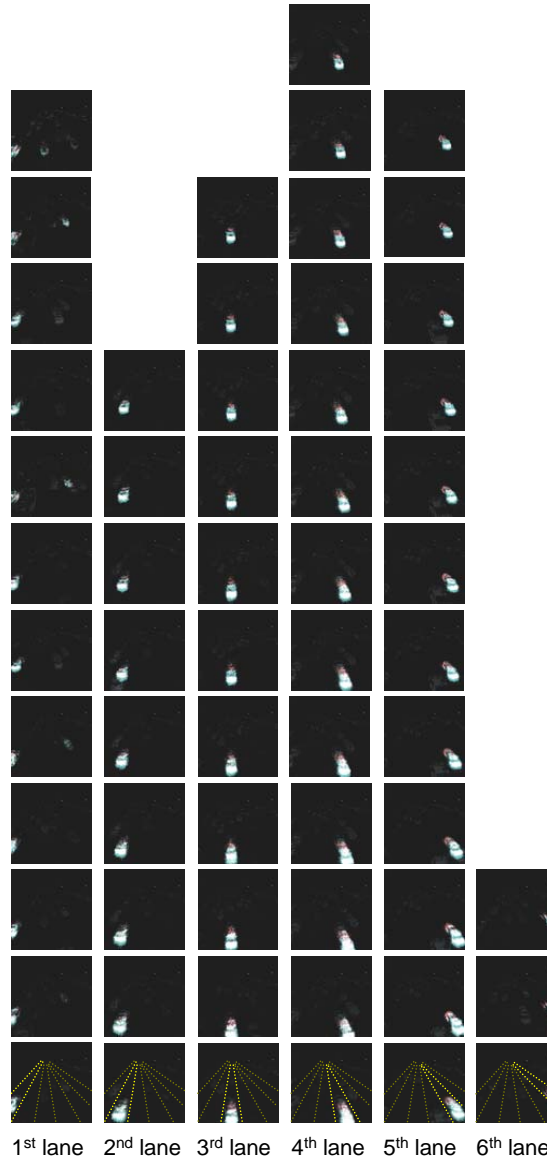


FIGURE 4.9 – Reconstruction des poids synaptiques de la première couche après apprentissage de la séquence des véhicules. Il y a 60 neurones et chacun d’eux est sélectif à un morceau de trajectoire spécifique pour seulement une voie de circulation. Le rouge représente les synapses renforcées connectées à la sortie positive (ON) des pixels et le bleu représente les synapses renforcées connectées à la sortie négative (OFF) des pixels. Lorsque les synapses ON et OFF sont toutes les deux renforcées pour un même pixel, la couleur résultante est gris clair.

de passage des véhicules sur les différentes voies de circulation, deux groupes de synapses appartenant spatialement à deux voies de circulation différentes ne peuvent pas rester tous les deux renforcés en moyenne. Grâce aux conditions initiales et à l’inhibition latérale, le neurone devient nécessairement de plus en plus sélectif à l’un des deux groupes. La LTP renforce alors l’un des groupes plus que l’autre, quelles que soient les autres synapses activées en même temps, qui subiront en moyenne une LTD parce qu’elles ne sont pas corrélées temporellement. Si le seuil est suffisamment haut pour autoriser une bonne sélectivité du neurone, les véhicules activant ce groupe de synapse seront au bout d’un moment suffisant pour déclencher le neurone la plupart du temps. Cela ne fonctionne que si la LTD est systématiquement appliquée aux synapses qui ne subissent pas une LTP, même celles qui n’ont pas reçu d’impulsion pré-synaptique. Ainsi, le mécanisme STDP classique, modélisé avec l’équation $\Delta w = f(t_{post} - t_{pre})$

échoue pour réaliser cette tâche, car il n'est pas possible avec cette règle simple de diminuer le poids des synapses dont l'instant d'activation n'est justement pas corrélé avec l'impulsion post-synaptique.

Une alternative pourrait être d'utiliser des synapses avec fuite pour obtenir le même effet avec la STDP classique. En effet, les synapses qui ne seraient pas continuellement renforcées lors de l'activation du neurone seraient constamment affaiblies grâce à la fuite. Dans ce cas, la sélectivité du neurone pour un motif particulier pourrait être maintenue tant que ce motif apparaît de manière régulière dans les stimuli d'entrée, émulant ainsi une mémoire à court terme. Cette approche est toutefois moins efficace à implémenter logiciellement et est difficile à obtenir avec les nano-composants synaptiques non volatiles. Elle introduit également une contrainte temporelle inutile entre la dynamique des stimuli d'entrée (durée temporelle et espacement entre les trajectoires) et la constante de temps de fuite des synapses.

En utilisant le même mécanisme, une seconde couche de neurones complètement connectée à la première est capable d'apprendre des trajectoires complètes à partir des trajectoires partielles extraites avec la première couche. Avec les paramètres appropriés (voir tableau 4.3), cette seconde couche peut identifier des voies de circulation complètes en recombinaison de trajectoires partielles. L'activité de sortie de cette couche peut être utilisée pour compter le nombre de véhicules passant sur chaque voie de circulation, comme le montre le tracé de l'activité de la figure 4.10. Le taux de détection va de 47% pour la première voie à 100% pour la cinquième.

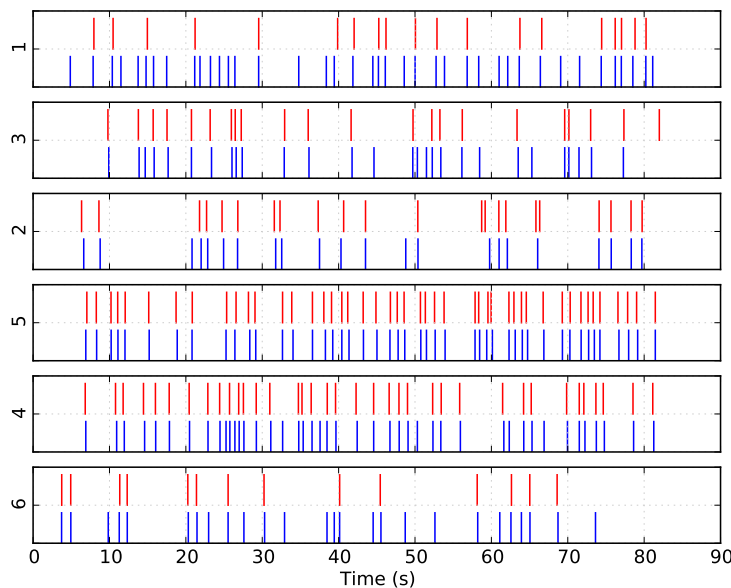


FIGURE 4.10 – Détection des véhicules sur chaque voie de circulation après apprentissage, avec la stratégie d'apprentissage globale. L'activité de référence, obtenue avec un étiquetage à la main, est comparée avec l'activité du meilleur neurone de la seconde couche pour la voie de circulation correspondante (numéroté de 1 à 6). L'activité de référence se situe en bas de chaque encart (en bleu) et l'activité en sortie du réseau est en haut (en rouge).

Le graphique de l'activité et la reconstruction des poids sont calculés après que la séquence AER d'entrée de 78.5 s ait été présentée 8 fois. Cela correspond à un temps réel d'apprentissage d'approximativement 10 minutes, après quoi l'évolution des poids synaptiques est faible. Il est notable qu'après seulement une présentation de la séquence, un début de spécialisation de la plupart des neurones est déjà apparent sur la reconstruction des poids et la majorité de trajectoires visibles à cette étape restent stables jusqu'à la fin de l'apprentissage.

TABLE 4.3 – Paramètres des neurones pour l'apprentissage de motifs complexes. Un jeu de paramètres différent est utilisé selon la stratégie d'apprentissage (global ou couche par couche).

Paramètre	Apprentissage global		Apprentissage couche par couche	
	1ère couche	2ème couche	1ère couche	2ème couche
I_{thres}	500000	1500	1060000	2240
T_{LTP}	12 ms	300 ms	14.7 ms	46.5 ms
T_{refrac}	300 ms	250 ms	517 ms	470 ms
$T_{inhibit}$	50 ms	100 ms	10.2 ms	182 ms
τ_{leak}	450 ms	300 ms	187 ms	477 ms
Taux recon.	47% to 100% / voie		98% en tout	

Apprentissage couche par couche

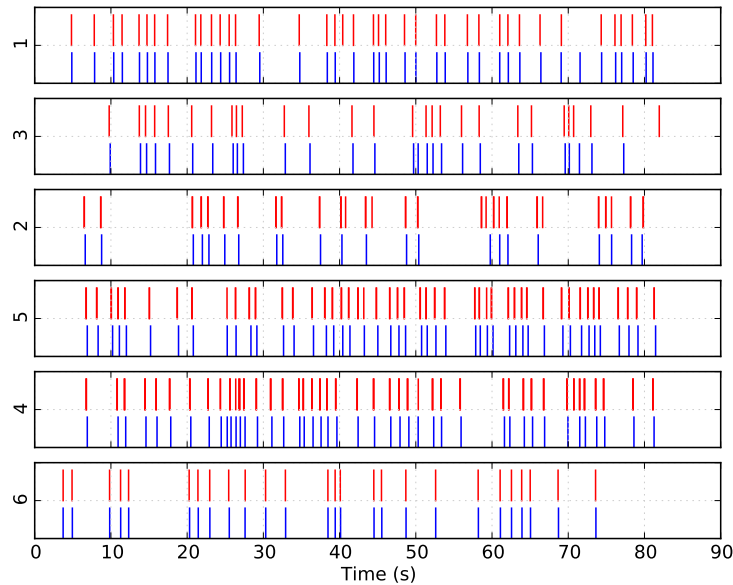


FIGURE 4.11 – Détection des véhicules sur chaque voie de circulation après apprentissage, avec la stratégie d'apprentissage optimisée couche par couche. L'activité de référence, obtenue par étiquetage manuel (en bleu) est comparée à l'activité du meilleur neurone de la seconde couche pour la voie de circulation correspondante (numéroté de 1 à 6) - en rouge.

Comme nous l'avons montré pour l'apprentissage de trajectoires partielles de balles, l'inhibition latérale n'est plus nécessaire lorsque les neurones sont spécialisés. En fait, l'inhibition latérale n'est même pas désirée, puisqu'elle peut empêcher des neurones légitimes de se déclencher lors de trajectoires se recouvrant temporellement. Cela n'empêche toutefois pas l'apprentissage à condition que la séquence soit suffisamment longue pour pouvoir considérer que les trajectoires à apprendre ne sont pas temporellement corrélées. Ce mécanisme, qui est fondamental pour permettre un apprentissage compétitif, est donc également la cause de performances médiocres en terme de détection de trajectoires après que l'apprentissage ait convergé. En conclusion, plus un neurone est sélectif, moins il a besoin d'être inhibé par ses voisins.

La figure 4.11 montre l'activité des neurones de sortie lorsque l'inhibition latérale et la STDP sont désactivées après l'apprentissage, sur la première couche, puis sur la seconde. La reconstruction des poids de la seconde couche est également représentée sur la figure 4.12. Celle-ci est obtenue en calculant la somme pondérée des reconstructions de la première couche, avec les poids synaptiques correspondant pour chaque neurone de la seconde couche. La durée en temps réel d'apprentissage est de 10 minutes par

couche, ce qui correspond à 20 minutes au total. Une détection quasiment exhaustive des trajectoires est possible. Le réseau apprend à détecter les véhicules passant sur chaque voie de circulation d'une manière totalement non supervisée, avec seulement 10 paramètres réglables en tout pour les neurones et sans avoir programmé le réseau de neurones pour effectuer une quelconque tâche. Nous sommes capables de compter les véhicules passant sur chaque voie de circulation avec une précision particulièrement bonne, simplement parce qu'il s'agit d'une conséquence de l'extraction de motifs temporellement corrélés. Sur 207 véhicules passant sur les six voies durant les 78.5 s de la séquence, seuls 4 véhicules sont ignorés, avec un total de 9 faux positifs, correspondant essentiellement à des camions activant les neurones deux fois ou des véhicules circulant au milieu de deux voies. Cela nous donne un taux de détection très bon de 98% sans même qu'un réglage fin des paramètres ne soit requis.

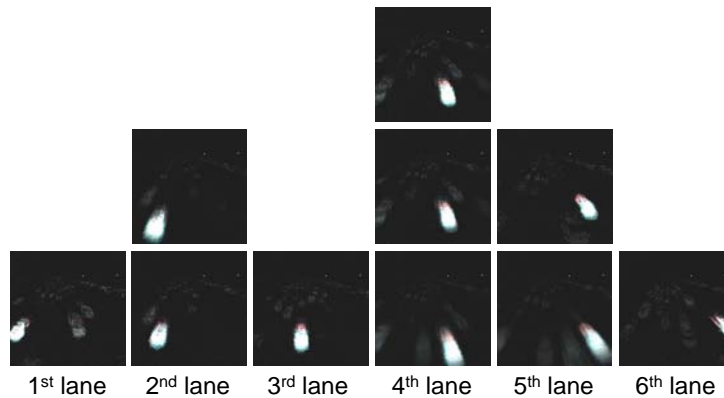


FIGURE 4.12 – Reconstruction des poids pour la seconde couche après l'apprentissage couche par couche (obtenue en calculant la somme pondérée des reconstructions de la première couche, avec les poids synaptiques correspondant pour chaque neurone de la seconde couche). Les neurones de la seconde couche associent plusieurs neurones de la première couche se déclenchant pour des successions de morceaux de trajectoire très rapprochées pour permettre une détection robuste de manière complètement non supervisée.

Si l'inhibition latérale est supprimée après apprentissage, mais que STDP reste active, nous avons observé que les motifs principaux extraits avec la première couche restent stables, comme cela était le cas pour l'apprentissage des trajectoires de balles. Toutefois, les neurones sensibles à la même voie de circulation finissent par apprendre le même morceau de trajectoire. Notre hypothèse est que la constante de temps de fuite est trop longue comparée à la période d'inhibition pendant l'apprentissage, ce qui signifie que même après apprentissage, l'inhibition latérale empêche les neurones de remonter les morceaux de trajectoires si STDP reste active.

Extraction spatialement localisée et combinaison de motifs

Dans la topologie du réseau précédent, la première couche est complètement connectée au capteur AER, résultant en un nombre total de synapses de 1 966 680. En effet, il n'était pas nécessaire de contraindre spatialement les entrées des neurones pour autoriser un apprentissage compétitif de motifs localisés. Pour des applications pratiques toutefois, il n'est pas très réaliste d'avoir pratiquement 30 000 synapses pour chaque motif à extraire (un par neurone). C'est pourquoi des approches plus hiérarchiques sont préférables (Wersing and Körner, 2003) et nous allons en introduire une dans ce qui suit. La rétine est divisée en 64 carrés de 8 par 8, de 16×16 pixels chacun, ce qui fait 64 groupes de neurones. L'inhibition latérale est restreinte à chaque groupe, comme illustré sur la figure 4.13. Avec 4 neurone par groupe, il y a au total $8 \times 8 \times 4 + 10 =$

266 neurones et $16 \times 16 \times 2 \times 8 \times 8 \times 4 + 8 \times 8 \times 10 = 131\,712$ synapses pour les couches du réseau.

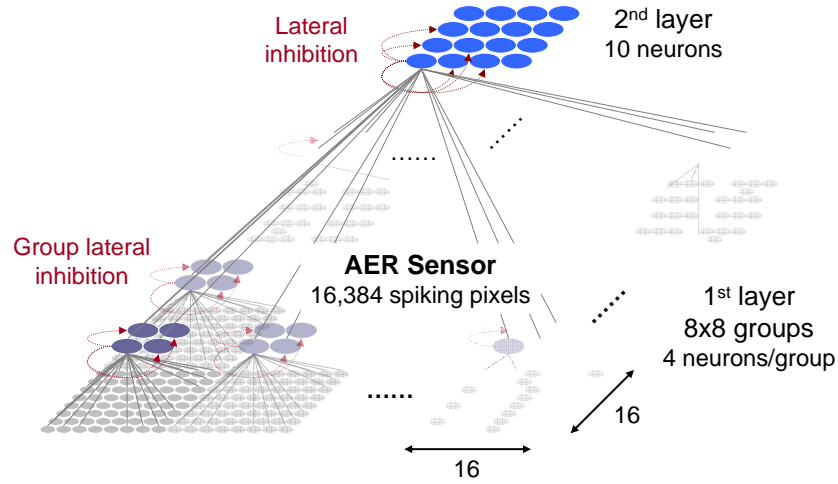


FIGURE 4.13 – Topologie du réseau de neurones pour l'apprentissage spatialement localisé, directement à partir de données enregistrées à partir d'un capteur AER. La première couche consiste en 64 groupes indépendants, chacun connecté à une partie différente du capteur avec 4 neurones par groupe, au lieu de neurones complètement connectés à la rétine.

En utilisant la méthode d'apprentissage couche par couche introduite précédemment, on obtient des taux de détections similaires ($> 95\%$ en tout), mais avec dix fois moins de synapses et un temps de simulation réduit (plus rapide que le temps réel) car il y a moins de neurones par pixel. Les paramètres d'apprentissage sont donnés dans la table 4.4. Tous les neurones des groupes de la première couche partagent les mêmes paramètres.

TABLE 4.4 – Paramètres des neurones pour l'extraction du sous-motifs (stratégie d'apprentissage couche par couche).

Paramètre	1ère couche	2ème couche
I_{thres}	60000	2000
T_{LTP}	6 ms	70 ms
T_{refrac}	700 ms	300 ms
$T_{inhibit}$	100 ms	450 ms
τ_{leak}	100 ms	250 ms
Taux recon.	$> 95\%$ en tout	

Quelques statistiques sur l'apprentissage sont également données dans le tableau 4.5, pour le système multi-groupes et le système complètement connecté de la partie précédente. Il montre que la fréquence de mise à jour des poids synaptiques (c'est-à-dire la fréquence post-synaptique) est de l'ordre de 0.1 Hz et la fréquence pré-synaptique moyenne est sous les 2 Hz. De plus, ces fréquences sont indépendantes de la taille du réseau. Les fréquences moyennes pour la deuxième couche sont également similaires, mais, à cause du nombre réduit de synapses pour la topologie multi-groupes, le nombre total d'évènements est plus de dix fois plus faible. Ces chiffres sont relativement proches de ce qui peut se trouver dans les réseaux de neurones biologiques. La faible fréquence des impulsions dans ce type de réseaux de neurones est remarquable, au regard de la tâche de détection complexe réalisée. Elle est un bon indicateur du potentiel de mise à l'échelle et de la haute efficacité de l'association de capteurs de vision dynamiques et de réseaux de neurones impulsifs pour ce type de tâche, comparé à l'analyse

synchrone de mouvement classique, image par image.

TABLE 4.5 – Statistiques d'apprentissage, pour la durée complète de l'apprentissage ($8 \times 85 = 680$ s), pour le nouveau réseau, comparé à la topologie complètement connectée. Les nombres d'évènements de lecture par synapse dépendent uniquement des stimuli et sont donc identiques, contrairement au nombre total d'évènement de lecture.

	/synapse (moy.)	/synapse/s	Total
1ère couche (multi-groupes)			
Evt. de lecture	1270	1.86	166M
Evt. de LTP	8.18	0.012	1.07M
Evt. de LTD	81.6	0.12	10.7M
1ère couche (complètement connecté)			
Evt. de lecture	1270	1.86	2.490M
Evt. de LTP	7.61	0.011	15M
Evt. de LTD	244	0.36	481M
2ème couche (complètement connecté)			
Evt. de lecture	309	0.45	185k
Evt. de LTP	4.94	0.0073	2.96k
Evt. de LTD	103	0.15	61.8k

Formation de champs récepteurs

Dans la dernière expérience qui suit, nous avons utilisé la même topologie de réseau pour extraire des champs récepteurs à partir d'une séquence de type «marche dans l'environnement»¹ (où contrairement aux expériences précédentes, l'observateur se déplace au lieu d'être immobile). La séquence ne contient aucun motif qui se répète continument à l'échelle globale, pourtant la plupart des neurones au niveau des groupes deviennent sélectifs à différentes formes, de la simple ligne verticale ou d'un rond à des motifs plus complexes, comme vu sur la figure 4.14. Ces formes rappellent des champs récepteurs observés en biologie dans la couche V1, ou générés par plusieurs algorithmes d'apprentissage automatique (Wimbauer et al., 1994).

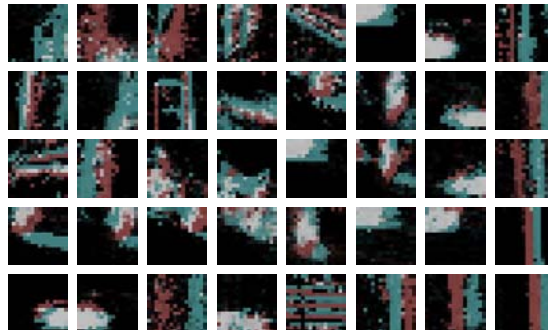


FIGURE 4.14 – Reconstructions des poids pour quelques neurones de la première couche, après l'apprentissage d'une séquence de type «marche dans l'environnement». Bien qu'il n'y ait pas de motif se répétant à l'échelle globale, des motifs locaux émergent.

1. Conduite dans Pasadena vers la poste. Disponible sur (DVS128 Dynamic Vision Sensor Silicon Retina data, 2011).

4.2.4 Robustesse et immunité au bruit

Dans cette partie, nous montrons que notre processus d'apprentissage est remarquablement tolérant à la variabilité synaptique, même lorsque de la variabilité au niveau des neurones est également ajoutée. Des paramètres neuronaux exacts ne sont donc pas requis. Cela montre que le réseau est robuste et ne nécessite pas un réglage fin de ses paramètres pour fonctionner (ce qui est également un critère d'évaluation pour une règle d'apprentissage «pratique»). Nous montrons également la tolérance extrême au bruit et au «jitter», à des niveaux bien supérieurs aux données déjà bruitées enregistrées à partir du capteur AER. En outre, l'analyse de la robustesse et de l'immunité au bruit est particulièrement importante si l'on considère une implémentation matérielle de la règle d'apprentissage proposée sur les nano-dispositifs synaptiques.

TABLE 4.6 – Statistiques du taux de détection sur 100 simulations, avec une dispersion de 20% pour tous les paramètres synaptiques. La dispersion est définie comme l'écart type par rapport à la valeur moyenne.

Voies apprises	Véhicules manqués [†]	Total (%)
	≤ 10	79
Cinq premières	> 10 et ≤ 20	10
	> 20	2
Seulement quatre	≤ 10	9
[†] Sur les voies apprises		100

TABLE 4.7 – Statistiques du taux de détection sur 100 simulations, avec une dispersion de 10% pour tous les paramètres neuronaux, en plus d'une dispersion de 20% pour tous les paramètres synaptiques.

Voies apprises	Véhicules manqués [†]	Total (%)
Toutes les six	≤ 20	1
Cinq premières	≤ 10	51
	> 10 et ≤ 20	21
	> 20	5
Cinq (autres)	≤ 10	3
	≤ 10	16
Seulement quatre	> 10 et ≤ 20	1
	> 20	2
[†] Sur les voies apprises		100

Variabilité synaptique

Le tableau 4.6 résume les résultats en terme de véhicules manqués pour un lot de 100 simulations, où une dispersion de 20% a été appliquée à l'ensemble des paramètres synaptiques : w_{min} , w_{max} , w_{init} , α_+ et α_- ($\beta_+ = \beta_- = 0$). Ceci est une variabilité considérable : 20% des synapses ont un poids maximum supérieur ou inférieur de 25% à la valeur maximale moyenne. Sur 100 simulations, 9 ont échoué à apprendre plus de 4 voies de circulation, mais même lorsque deux voies ne sont pas apprises, le taux de détection sur les autres reste supérieur à 95%. Il n'y a jamais eu plus de 10 faux positifs en tout sur les voies apprises. La sixième voie de circulation n'est jamais apprise. Il s'agit de la voie la plus difficile à identifier, car les véhicules passant sur celle-ci (tout à droite de la rétine) activent moins de pixels au cours de leur trajectoire que sur les autres voies et le nombre total de véhicules passant sur cette voie est également plus faible. En

conséquence, comme l'activité impulsionnelle sur la voie 6 est au moins 50% plus faible que sur les autres, il est vraisemblable qu'en fonction des conditions initiales ou une valeur critique de certains paramètres, aucun neurone ne soit capable de suffisamment renforcer les synapses correspondantes pour devenir exclusivement sélectif à cette voie. En effet, la figure 4.9 montre un exemple spécifique où toutes les voies de circulation ont été apprises et seuls 3 neurones sur 60 parviennent à devenir sélectifs à la dernière voie.

Variabilité neuronale

Un nouveau lot de 100 simulations a été effectué, cette fois-ci avec une dispersion supplémentaire de 10% appliquée à tous les paramètres neuronaux du tableau 4.3, pour les deux couches. Les résultats du tableau 4.7 sont encore bons pour 75% des simulations et très bons pour 50% d'entre elles, si on ignore la sixième voie. Dans les pires cas, quatre voies sont apprises. Il est notable que les voies 4 et 5 sont toujours correctement apprises, avec toujours plus de 90% des véhicules détectés dans toutes les simulations. Cela est dû au fait que ces voies sont facilement identifiables (contrairement aux voies 1 et 6) et subissent le plus de trafic (le double que sur les voies 2 et 3). De meilleurs résultats pourraient être possibles avec des séquences AER plus longues, sans même considérer la possibilité d'augmenter la résolution du capteur.

Bruit et «jitter»

La robustesse au bruit et au «jitter» du mécanisme d'apprentissage proposé a également été évaluée. Du bruit blanc ajouté à la séquence (de manière à ce que 50% des impulsions de la séquence soient aléatoires) et 5 ms de «jitter» ajouté n'ont eu pratiquement aucun impact sur l'apprentissage de la séquence. Bien que seules les cinq premières voies de circulation soient apprises, essentiellement pour les raisons décrites précédemment, il y a moins de 5 véhicules manqués et 10 faux positifs avec les paramètres du tableau 4.3. Cela montre encore une fois qu'un système basé sur STDP fonctionne sur des données extrêmement bruitées sans nécessiter de filtrage particulier.

4.3 Détection de motifs auditifs dans du bruit

Dans cette partie, nous montrons comment la méthodologie et les résultats obtenus pour l'apprentissage de motifs visuels peuvent être transposés pour l'apprentissage de motifs auditifs. De même que pour la rétine, le codage de l'information dans l'oreille humaine, qui a lieu dans la cochlée, est très différent de l'échantillonnage classique utilisé en traitement du signal. Les algorithmes de reconnaissance de la parole se rapprochent en revanche, au niveau du pré-traitement, du fonctionnement de la cochlée. Typiquement, le pré-traitement consiste à extraire des tranches de quelques dizaines de millisecondes de signal et y appliquer une transformée de Fourier pour obtenir un vecteur codant l'amplitude du signal à plusieurs fréquences. Ce vecteur est alors comparé à un ensemble de vecteurs prototypes, ou plus généralement à un modèle statistique dans le cas des chaînes de modèle de Markov caché – «Hidden Markov Model» (HMM), pour identifier les mots et la parole.

La construction de ces vecteurs prototypes se fait par apprentissage et les modèles de réseaux de neurones classiques se sont déjà révélés très performants pour l'extraction de motifs auditifs de bas niveau, utilisés en amont de HMM (Bengio et al., 1992). Les réseaux de neurones statiques ne permettent toutefois pas de modéliser de manière satisfaisante des motifs complexes contenant des dépendances temporelles (Bengio et al., 1992). Les motifs de bas niveau sont en effet extraits à partir de tranches de signal de

courte durée (10 ms), qui sont supposées quasi-stationnaires. Des réseaux impulsionnels, exploitant le codage temporel de l'information, pourraient lever cette limitation. En outre, une implémentation memristive efficace éliminerait le goulot d'étranglement lié à l'accès aux données.

Pour démontrer le potentiel d'une telle implémentation, nous proposons un modèle simple constitué d'un unique neurone équipé d'un apprentissage de type STDP. Nous montrons que celui-ci est capable d'expliquer la formation rapide de la mémoire auditive chez l'être humain pour du bruit blanc. Nous nous référons pour cela aux expériences psychoacoustiques de (Agus et al., 2010), qui sont parmi les premières à mettre en évidence cette capacité à apprendre du bruit, et montrent que la mémoire ainsi acquise peut durer des semaines. Avec ce modèle, nous obtenons des résultats en termes de capacité et temps d'apprentissage compatibles avec ceux observés lors de ces expériences psychoacoustiques. Celui-ci s'avère même capable d'expliquer certaines invariances en détection non triviales, tel que l'invariance à la direction du temps du motif audio.

4.3.1 Méthodologie

Nous simulons un neurone impulsif faisant de l'extraction d'un motif répétitif dans du bruit. Pour cela, nous utilisons le simulateur événementiel C++ Xnet pour l'ensemble des simulations. Une boîte à outils complète pour le traitement de signal a été développée et associée à Xnet pour ces travaux : lecture/écriture de fichiers de type «WAVEform audio file format» (WAV), filtrage, ré-échantillonnage, transformée de Fourier rapide – «Fast Fourier Transform» (FFT), spectrogramme...

Règle d'apprentissage et modèle du neurone

La règle d'apprentissage est la règle STDP simplifiée déjà introduite dans la partie précédente. Le modèle d'évolution des poids synaptiques est identique également, mais dans les simulations qui suivent, contrairement à l'apprentissage des données AER, les paramètres β sont non nuls, $\beta_+ = \beta_- = 3$. Cela a pour effet de simuler une règle d'évolution des poids pseudo-multiplicative. Cela a une conséquence importante pour l'apprentissage, qui sera mise en évidence dans la dernière partie de ce chapitre. Le neurone est de type LIF et est modélisé de la même manière que pour la détection de motifs visuels.

Modélisation de la cochlée

La cochlée est modélisée de manière très classique et la plus simple possible par un banc de filtres passe-bande, suivis de détecteur d'enveloppe, comme illustré sur la figure 4.15. Les filtres sont de type Butterworth d'ordre 8. La bande passante des filtres est déterminée en utilisant l'échelle bande passante rectangulaire équivalente – «Equivalent Rectangular Bandwidth» (ERB), approximée par la fonction suivante (Glasberg and Moore, 1990), avec f la fréquence centrale du filtre et $ERB(f)$ la bande passante correspondante :

$$ERB(f) = f/Q_{ear} + BW_{min} \text{ avec } Q_{ear} = 9.26449 \text{ et } BW_{min} = 24.7 \quad (4.5)$$

Q_{ear} est le coefficient de qualité du filtre asymptotique aux hautes fréquences et BW_{min} est la bande passante minimale pour les canaux de basse fréquence. L'espacement des fréquences centrales suit une loi géométrique d'après l'équation suivante (Slaney, 1993) :

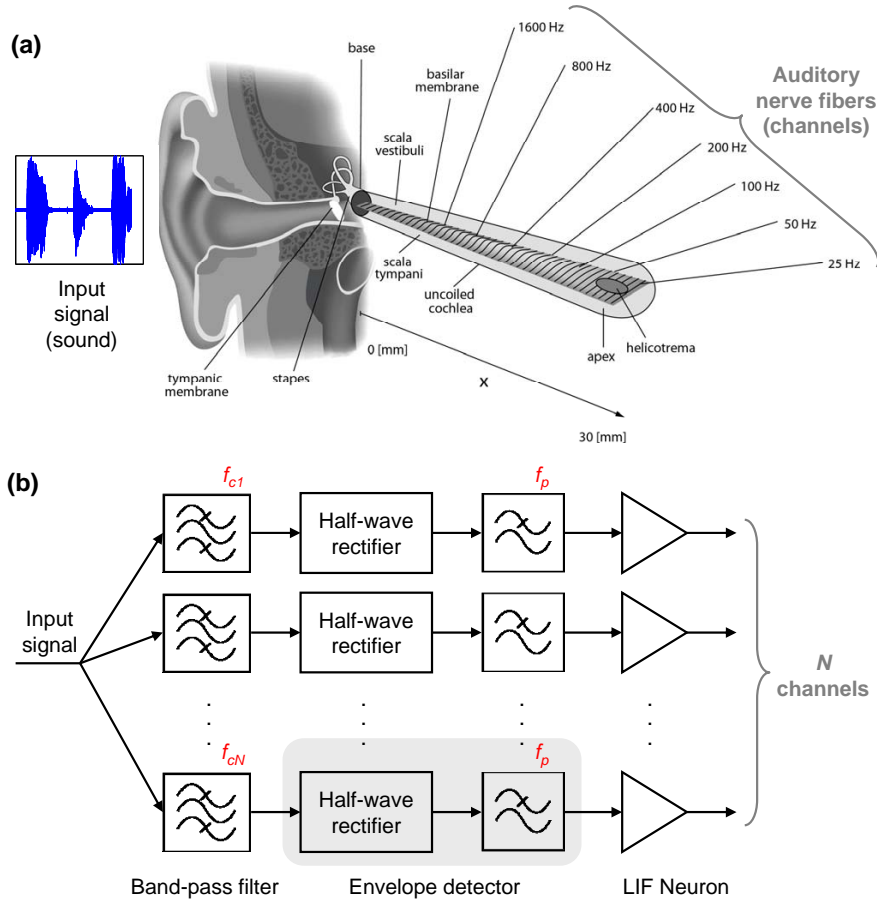


FIGURE 4.15 – Modélisation de la cochlée humaine. (a) Système auditif humain avec une vue déroulée de la cochlée, montrant l'évolution de la sensibilité en fréquence des fibres nerveuses en fonction de leur position le long de celle-ci. (b) Modélisation de la cochlée. Chaque canal est sensible à une plage de fréquence. Le signal est d'abord filtré dans un filtre passe-bande, puis un redresseur suivi d'un filtre passe-bas en extrait l'enveloppe. Un neurone LIF émule pour chaque canal la transduction du signal en impulsions et modélise ainsi la fibre nerveuse en sortie du canal.

$$f_{ci}|_{i=0..N-1} = -Q_{ear} \cdot BW_{min} + (f_{low} + Q_{ear} \cdot BW_{min}) \cdot \exp\left(\frac{i}{N} \cdot [\log(f_{up} + Q_{ear} \cdot BW_{min}) - \log(f_{low} + Q_{ear} \cdot BW_{min})]\right) \quad (4.6)$$

L'évolution du gain d'un filtre en fonction de sa fréquence centrale sur l'échelle ERB est représentée sur la figure 4.16. Les fréquences minimales et maximales du banc de filtres sont respectivement $f_{low} = 50$ Hz et $f_{up} = 16$ KHz. Le filtre passe-bas du détecteur d'enveloppe est un filtre de Butterworth d'ordre 1 et de fréquence de coupure 65 Hz.

Génération des stimuli

Les stimuli utilisés pour l'apprentissage sont constitués de tranches de 1 s de bruit blanc. Celles-ci sont de deux types : les tranches N, qui sont toutes différentes et générées aléatoirement et les tranches RefRN, qui sont toutes identiques dans un même stimulus et qui constituent donc le motif répétitif à extraire. Une tranche RefRN correspond également à 1 s de bruit blanc et est statistiquement indistinguable d'une tranche N. En outre, il n'y a jamais deux tranches de types RefRN qui se suivent, celles-ci sont insérées

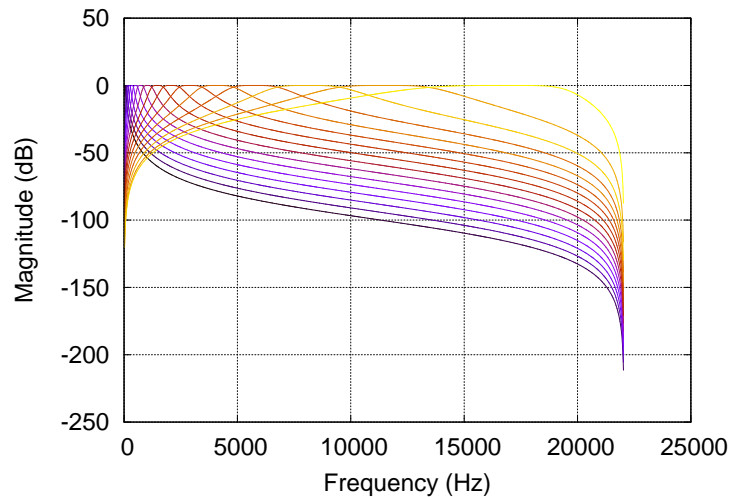


FIGURE 4.16 – Mise en évidence de l'échelle ERB pour les filtres passe-bande constituant le modèle de la cochlée. Les caractéristiques de 16 filtres entre la fréquence minimale et la fréquence maximale sont représentés. La sensibilité est plus grande dans les basses fréquences.

aléatoirement entre des tranches N. L'extraction des instants d'occurrence des tranches RefRN nécessite un apprentissage et une mémorisation, puisque leur identification ne peut se baser que sur leur répétition dans le stimulus.

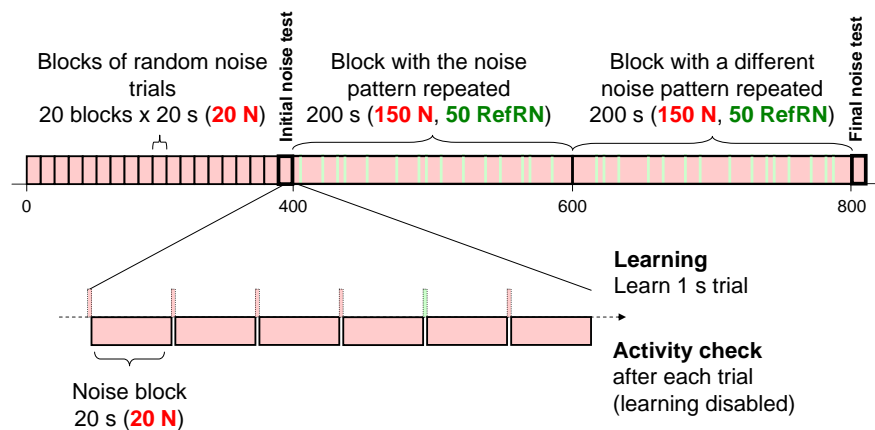


FIGURE 4.17 – Construction d'un stimulus auditif pour l'extraction d'un motif répétitif (RefRN). Les premières 400 s sont constituées de bruit blanc, suivi du bloc contenant le motif RefRN, d'une durée de 200 s. Un bloc témoin de 200 s avec un autre motif RefRN suivi d'un bloc de bruit de 20 s permettent d'évaluer la sélectivité du neurone après l'apprentissage (en haut). Durant la présentation du stimulus, un bloc de 20 s de bruit teste le taux d'activité du neurone après chaque tranche N ou RefRN (de durée 1 s).

Pour générer une tranche de bruit, un signal échantillonné à 44.1 KHz est construit à partir de tirages aléatoires d'une loi normale de moyenne nulle et d'écart-type l'unité. Les tranches de 1 s sont ensuite accolées pour former un stimulus d'une durée totale de 820 s, de la manière détaillée sur le haut de la figure 4.17. Les blocs de bruit de 20 s représentés sur la figure, composés de 20 tranches N, sont tirés aléatoirement d'une bibliothèque de 200 blocs générée à l'avance pour l'ensemble des stimuli pour une question de temps de calcul.

Les trains d'impulsions obtenus en sortie de notre modèle de cochlée pour une tranche N ou RefRN sont présentés sur la figure 4.18. Le train d'impulsions en sortie de chaque canal de la cochlée est envoyé sur une synapse d'entrée du neurone.

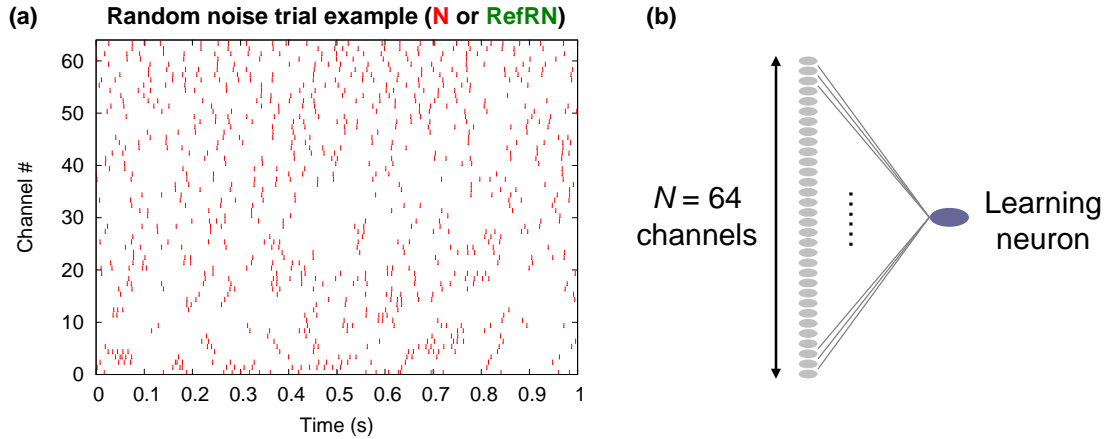


FIGURE 4.18 – (a) Trains d’impulsions générés par notre modèle de cochlée avec 64 canaux, pour une tranche de bruit blanc. (b) Topologie du réseau : un seul neurone équipé de STDP est entièrement connecté aux canaux de sortie de notre modèle de cochlée.

4.3.2 Résultats des simulations

La figure 4.19 présente l’évolution de l’activité du neurone durant l’apprentissage d’un stimulus tel que décrit à la figure 4.17. L’apprentissage se fait sur la durée du stimulus, qui n’est présenté qu’une seule fois en entrée du système. L’apprentissage STDP est actif durant toute la durée de la séquence. Le premier résultat remarquable, en 4.19(a) est qu’en l’absence de motif répétitif (sur les 400 premières secondes du stimulus), l’activité du neurone se maintient à un peu plus d’un événement par seconde en moyenne. Cela est le résultat de l’utilisation d’une règle d’évolution des poids pseudo-multiplicative, qui permet de conserver une valeur moyenne des poids synaptiques stable avec STDP (Querlioz et al., 2011b). En 4.19(b), l’activité du neurone lors des premières présentations du motif RefRN est enregistrée. Le neurone n’est initialement pas spécialement sélectif à la tranche RefRN, puisqu’il continue de se déclencher pour les tranches N. En 4.19(c), l’activité du neurone est à nouveau enregistrée après 2 min de présentation du bloc contenant les tranches RefRN. Celui-ci est devenu complètement sélectif au motif répétitif et ne se déclenche plus pour le bruit N, après seulement une vingtaine de répétitions de RefRN. Cette sélectivité est vérifiée en 4.19(d), avec une activité nulle pour 20 s de bruit blanc.

Sélectivité après apprentissage

L’évolution de la sélectivité du neurone au cours de l’apprentissage est illustrée sur la figure 4.20. Celle-ci met clairement en évidence le maintien de l’activité du neurone avant l’apparition d’un motif répétitif (entre 0 et 400 s). Le neurone devient très rapidement sélectif aux tranches RefRN, après seulement 10 à 20 apparitions. Ces chiffres sont remarquablement similaires aux résultats des expériences psychoacoustiques (Agus et al., 2010). La probabilité du neurone de se déclencher sur une tranche de bruit après apprentissage (au-delà de 600 s) avoisine dans le cas présent les 0.1% (5 activations sur un total cumulé de 4400 s). Cette probabilité tombe facilement à 0 sur des simulations impliquant 3000 canaux auditifs au lieu de 64. Dans ce cas, le neurone devient véritablement une « cellule grand-mère », c’est-à-dire qu’il est sélectif à un motif et sa probabilité de se déclencher sur d’autres motifs aléatoires est extrêmement faible (Bowers, 2009).

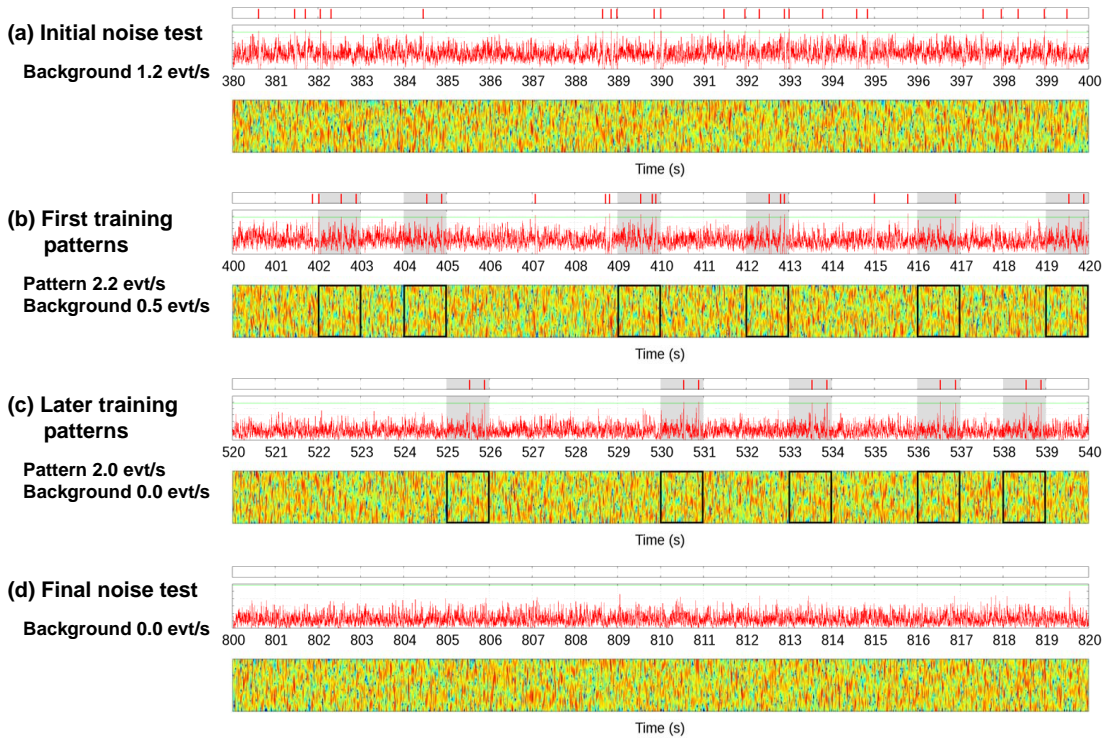


FIGURE 4.19 – Apparition de sélectivité au motif de bruit blanc audio répétitif avec STDP. Pour chaque extrait de l'activité du neurone, le potentiel de membrane (en haut) ainsi que le spectrogramme du stimulus (en bas) sont représentés. Les occurrences du motif répétitif RefRN sont mise en avant par un rectangle gris. Au fur et à mesure du déroulement de la séquence dans le temps, le neurone devient exclusivement sélectif au motif RefRN.

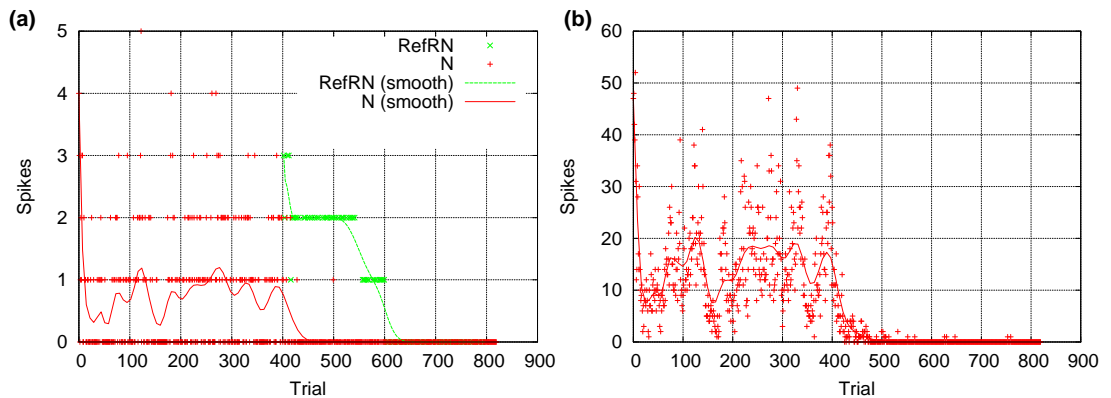


FIGURE 4.20 – Évolution de la sélectivité du neurone durant l'apprentissage. (a) Nombre d'activations du neurone par tranche de 1 s, qui peut être de type N (en rouge) ou RefRN (en vert). (b) Mesure de l'activité du neurone pour 20 s de bruit blanc au fur et à mesure de la présentation des tranches du stimulus. L'état du neurone est sauvegardé avant la mesure puis restauré après les 20 s de bruit et STDP est désactivé durant ce test.

Sensibilité et reproductibilité

La sensibilité (d') est une mesure couramment utilisée dans la théorie de détection du signal. Formellement, elle donne une mesure de la séparation entre la moyenne de la distribution obtenue lorsque le signal est présent et la moyenne de la distribution obtenue en l'absence de signal (bruit), pour une grandeur de sortie d'un capteur. La sensibilité s'exprime en écart-type de la distribution du bruit. La sensibilité peut s'exprimer en fonction du taux de succès («hit-rate») et du taux de fausses alarmes

(«false-alarm rate») (Macmillan and Creelman, 2004) :

$$d' = Z(\text{hit rate}) - Z(\text{false alarm rate}) \quad (4.7)$$

avec $Z(x)$ la bijection réciproque de la fonction de répartition de la loi normale.

L'évolution de la sensibilité moyenne, mesurée entre 400 et 600 s (correspondant à la séquence contenant les tranches RefRN à identifier) pour 10 stimuli différents, est représentée sur la figure 4.21, avec les barres d'erreurs. À droite de la figure, les activités moyennes mesurées pour les 10 stimuli sont superposées. La sensibilité moyenne aux tranches RefRN atteint les 2.7 à la fin de l'apprentissage, ce qui est comparable et même supérieur à celle mesurée en psychoacoustique avec un protocole expérimental identique, allant en moyenne de 2.0 à 2.7 selon les stimuli (Agus et al., 2010).

En répétant l'expérience sur plusieurs stimuli, il est notable que certains ne sont pas toujours appris. Si la même observation a été faite lors des expériences psychoacoustiques, qui ont mis en évidence certains stimuli plus difficiles à apprendre que d'autres, il semblerait que dans le cas de nos simulations la capacité à apprendre un stimuli soit fortement liée aux conditions initiales. La répétition de l'expérience pour un même stimulus le prouve, avec un taux d'échec pouvant varier entre 0 et 20-30% sur une dizaine d'essais avec des conditions initiales variables.

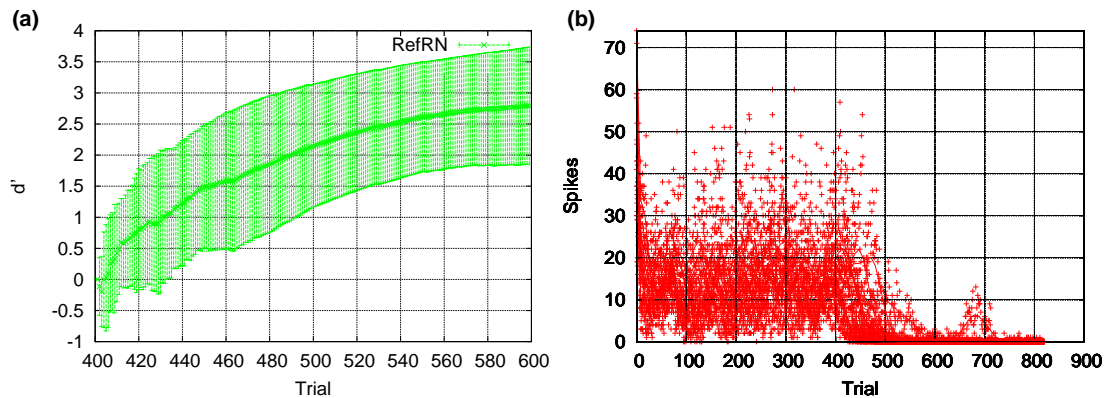


FIGURE 4.21 – Statistiques de l'évolution de la sensibilité moyenne du neurone durant l'apprentissage pour 10 stimuli différents. (a) Moyenne et écart-type de la mesure de sensibilité d' («d-prime») au motif répétitif du stimulus. (b) Mesure de l'activité moyenne, sur les 10 stimuli, du neurone pour 20 s de bruit blanc au fur et à mesure de la présentation des tranches du stimulus.

Bilan et perspectives avec les neurosciences

Nous avons introduit un modèle extrêmement simple de système auditif, puisqu'il fait appel à une modélisation rudimentaire de la cochlée et ne comporte qu'un neurone connecté à ses 64 canaux. Avec seulement 64 synapses équipées de la règle d'apprentissage STDP simplifiée introduite dans la partie précédente, nous avons montré que le système est capable d'extraire de manière non supervisée le motif répétitif dans une séquence de bruit blanc, le motif seul étant indiscernable du reste de la séquence. Ce modèle reproduit étonnamment bien l'évolution de la sensibilité et la vitesse d'apprentissage observée chez l'être humain.

D'autres observations devraient pouvoir être reproduites avec notre modèle, comme l'invariance à la direction du temps du motif audio, ou la tolérance à la compression du temps. Celles-ci font l'objet de travaux en cours et ne seront pas détaillées ici. Enfin, l'application à la reconnaissance de motifs significatifs, voire à la reconnaissance de parole, devrait pouvoir découler naturellement de ces résultats.

4.4 Apprentissage et classification de motifs

Les applications d'extraction de motifs temporels visuels ou auditifs introduites jusqu'à présent visent à identifier des paquets d'évènements fortement corrélés temporellement et à renforcer le poids des synapses associées à ces évènements. Si l'algorithme d'apprentissage STDP s'avère très performant pour ce type de tâches, celles-ci ne constituent pas des problèmes de classification. En effet, l'extraction de motif consiste à identifier le support spatial de groupes d'évènements corrélés, tandis que la classification nécessite d'identifier les corrélations entre les motifs d'entrée de même nature pour construire un prototype mémorisé dans les poids synaptiques.

Dans cette partie, nous essayons de répondre à deux problématiques. La première est de savoir si ces deux problèmes, extraction de motifs répétitifs et classification, sont équivalents pour l'algorithme d'apprentissage STDP. La deuxième, le cas échéant, est de savoir si STDP peut offrir des performances en classification comparables aux algorithmes Hebbiens classiques, tels que la règle du Delta. Pour se faire, nous présenterons une formalisation partielle de notre règle STDP simplifiée, ainsi qu'un exemple d'apprentissage et de classification sur la base de chiffres manuscrits MNIST. Sur cet exemple, nous obtenons des performances en reconnaissance comparables à un perceptron classique de même taille.

4.4.1 Méthodologie

Génération des stimuli

Pour de la classification, les stimuli sont générés à partir d'images statiques. Si l'on reste dans le domaine spatial, comme c'est le cas dans toutes nos simulations, à chaque pixel de l'image est associée une entrée du réseau, via un neurone d'entrée. La fonction du neurone d'entrée est de convertir la valeur du pixel en une ou plusieurs impulsions codant celle-ci dans le temps. Le codage peut être fréquentiel ou basé sur l'ordre (Van Rullen and Thorpe, 2001). Dans le cas d'un codage fréquentiel, STDP va avoir tendance à renforcer le poids des synapses activées les plus fréquemment (voir la formalisation dans la suite). La figure 4.22 représente plusieurs variations possibles pour un codage fréquentiel. Un codage basé sur l'ordre a l'avantage de ne nécessiter qu'un seul évènement par synapse, dont l'instant d'arrivée (ou «rang») est proportionnel à la valeur «informative» du pixel. Dans ce cas, STDP va avoir tendance à renforcer le poids des synapses activées en premier (Guyonneau et al., 2005).

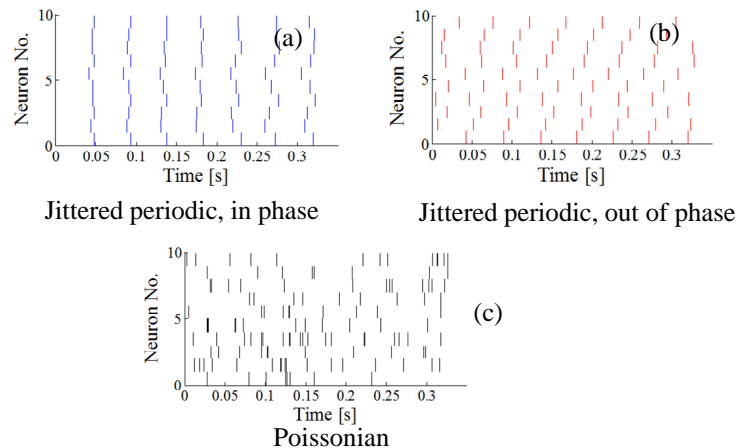


FIGURE 4.22 – Illustration de trois variations possibles d'un codage fréquentiel d'une image statique : (a), (b) et (c). Pour chaque codage, les évènements générés par dix neurones d'entrée associées à des pixels de même valeur sont représentés.

Ce sont les synapses associées aux pixels les plus informatifs, ou discriminatifs, pour l'image à apprendre qui doivent être renforcées. Pour que cela se produise avec STDP, les pixels informatifs doivent donc être codés avec une fréquence élevée, ou avec un rang faible. Les pixels les plus discriminatifs ne sont pas nécessairement ceux de luminance la plus élevée dans une image. Sur une photo par exemple, la partie la plus lumineuse est souvent le ciel, qui ne contient aucune information sur l'image. Si l'on prend exemple sur le fonctionnement de la rétine, ce sont essentiellement les saillances dans l'image qui produisent une réponse des cellules ganglionnaires. Comme ce sont souvent les contours qui sont les plus discriminatifs, ce type de réponse est adapté pour un apprentissage STDP. Pour émuler cette caractéristique de la rétine, des filtres d'extraction de contours sont classiquement utilisés. Le plus élémentaire est le filtre de Laplace 3×3 . Celui-ci peut être à centre positif ou négatif, émulant ainsi les cellules ganglionnaires à centre ON ou à centre OFF :

$$L_+ = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad L_- = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (4.8)$$

D'autres noyaux de convolution sont utilisés, tels que les noyaux de Gabor (Daugman, 1985), qui donnent l'orientation des contours ou des mouvements, émulant ainsi une partie des champs récepteurs de la couche V1 du système visuel humain. L'information de luminance après convolution peut ensuite être utilisée pour coder la fréquence ou le rang, par exemple avec une relation proportionnelle (Thorpe et al., 2010).

En ce qui concerne la résolution nécessaire pour classifier des images, les expériences psychovisuelles montrent que des objets et des scènes sont reconnaissables avec des images de résolution aussi basse que 32×32 pixels (Torralba, 2009). En outre, les informations de colorimétrie n'améliorent pas significativement les taux de reconnaissance (Delorme et al., 2000).

Formalisation de la règle d'apprentissage

Dans cette section, nous proposons une formalisation partielle de la règle d'apprentissage STDP simplifiée dans un cas simple, qui permet néanmoins d'interpréter un certain nombre de résultats sur l'évolution des poids synaptiques dans nos différentes simulations. Nous verrons également les conséquences sur le codage de l'information.

Pour ce faire, nous cherchons dans un premier temps une relation entre la probabilité d'une synapse de subir une LTP et l'évolution de son poids moyen. On note $P(LTP)$ la probabilité d'une synapse d'être renforcée et $P(LTD)$ la probabilité d'une synapse d'être affaiblie. Avec notre règle d'apprentissage STDP, on a :

$$P(LTD) = 1 - P(LTP) \quad (4.9)$$

Dans un souci de simplification, on suppose que le poids des synapses w est compris entre 0 et 1. Notre modèle d'évolution des poids est donc le suivant :

$$w \in [0, 1] \quad (4.10a)$$

$$\Delta w_+(w) = \Delta_+ \cdot \exp^{-\beta \cdot w} \quad (4.10b)$$

$$\Delta w_-(w) = -\Delta_- \cdot \exp^{-\beta \cdot (1-w)} \quad (4.10c)$$

avec $\Delta w_+(w)$ l'augmentation du poids synaptique induit par une LTP et $\Delta w_-(w)$ la diminution du poids synaptique induite par une LTD. La suite de l'évolution du poids moyen d'une synapse peut s'écrire de la manière suivante :

$$\overline{w}_{n+1} = \overline{w}_n + \Delta w_+(\overline{w}_n) \cdot P(LTP) + \Delta w_-(\overline{w}_n) \cdot P(LTD) \quad (4.11)$$

En calculant le point fixe de cette suite, on obtient l'expression du poids moyen de la synapse en régime permanent $\overline{w} = \overline{w}_{n+1} = \overline{w}_n$:

$$\begin{aligned} \Delta w_+(\overline{w}) \cdot P(LTP) &= -\Delta w_-(\overline{w}) \cdot P(LTD) = -\Delta w_-(\overline{w}) \cdot (1 - P(LTP)) \\ \Delta_+ \cdot \exp^{-\beta \cdot \overline{w}} \cdot P(LTP) &= \Delta_- \cdot \exp^{-\beta \cdot (1-\overline{w})} \cdot (1 - P(LTP)) \\ \overline{w} &= \frac{1}{2} + \frac{1}{2\beta} \cdot \ln \left(\frac{\Delta_+ \cdot P(LTP)}{\Delta_- \cdot (1 - P(LTP))} \right) \end{aligned} \quad (4.12a)$$

Dans le cas où $\beta_+ \neq \beta_-$, l'expression devient :

$$\overline{w} = \frac{\beta_-}{\beta_+ + \beta_-} + \frac{1}{\beta_+ + \beta_-} \cdot \ln \left(\frac{\Delta_+ \cdot P(LTP)}{\Delta_- \cdot (1 - P(LTP))} \right) \quad (4.13)$$

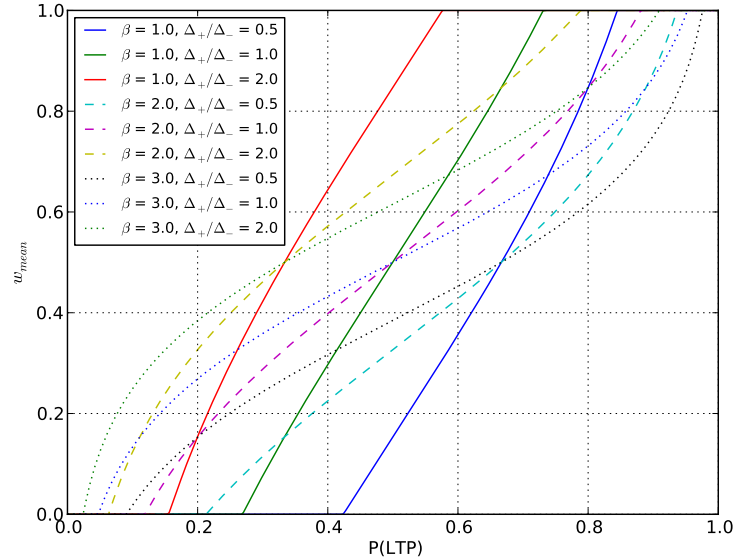


FIGURE 4.23 – Poids synaptique moyen \overline{w} en fonction de la probabilité de LTP pour différents jeux de paramètres. β règle la pente au point d'inflexion et le rapport Δ_+/Δ_- règle le décalage de la plage de variation du poids par rapport à $P(LTP)$.

L'équation ci-dessus donne donc une relation formelle entre le poids moyen d'une synapse et sa probabilité d'être renforcée. Celle-ci est représentée sur la figure 4.23. Elle montre que STDP peut maintenir des poids intermédiaires stables pour certaines valeurs de $P(LTP)$, à condition que le modèle de mise à jour des poids synaptiques soit pseudo-multiplicatif. Dans le cas limite où $\beta \rightarrow 0$, on se retrouve dans le cas purement additif et le tracé de la caractéristique correspondante sur la figure 4.24 montre que dans ce cas l'apprentissage STDP ne peut résulter que sur des valeurs de poids binaires. Ceci est effectivement observé à la fin de l'apprentissage dans l'application d'extraction de motifs visuels, pour laquelle β avait été laissé à zéro. Une deuxième conséquence importante de ce résultat est que l'activité d'un neurone ne peut pas être maintenue avec des poids additifs si les probabilités de LTP des synapses sont identiques. En effet, dans le cas où $P(LTP)$ est le même pour toutes les synapses, l'ensemble des poids synaptiques va converger vers 0 (si $P(LTP) < 0.5$, ce qui est généralement le cas pour que le neurone puisse devenir sélectif). Ceci est le cas de l'application d'extraction de motif auditif, où des poids pseudo-multiplicatifs ont été utilisés pour que le neurone

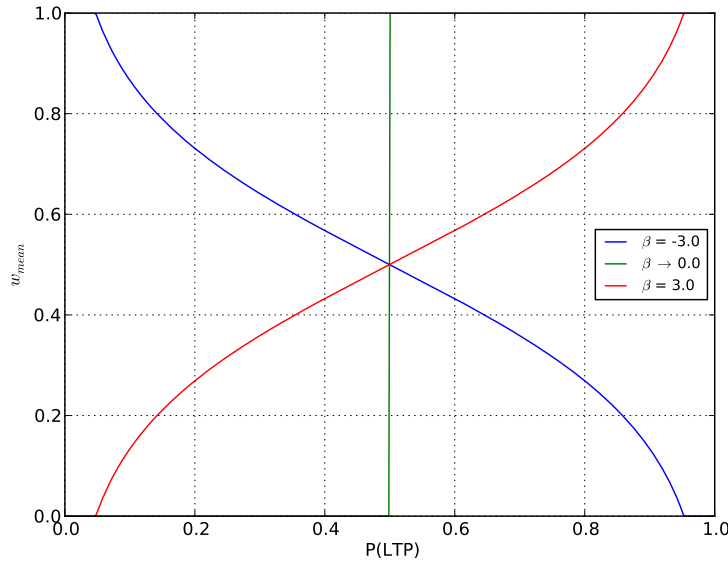


FIGURE 4.24 – Cas limites du poids synaptique moyen \bar{w} en fonction de la probabilité de LTP pour $\beta = 0, 3$ et -3 .

puisse maintenir une activité moyenne sur du bruit blanc et permettre au système de fonctionner.

Dans le cas d'un codage fréquentiel, il est aisé de mettre en relation $P(LTP)$ avec la fréquence en entrée de la synapse $f(x)$, x étant la valeur du pixel associé (que l'on suppose normalisée entre 0 et 1). En notant T_{LTP} la fenêtre de LTP de STDP, on a simplement :

$$P(LTP) = \begin{cases} f(x).T_{LTP} & \text{si } f(x) < T_{LTP} \\ 1 & \text{sinon} \end{cases} \quad (4.14)$$

Ainsi, dans le cas simple d'une relation linéaire de type $f(x) = f_{max}.x$ (avec $x \in [0, 1]$), et si l'on suppose que $f_{max} = 1/T_{LTP}$, on obtient $P(LTP) = x$. Si l'on considère désormais l'ensemble I des images pour lesquelles un même neurone se déclenche, en supposant que chaque image est équiprobable, on obtient globalement $P(LTP) = \bar{x}$, avec \bar{x} la valeur du pixel de l'image moyenne \bar{I} .

De cette étude nous concluons que, sous réserve des hypothèses émises dans cette partie, un neurone équipé de STDP mémorise dans ses poids synaptiques une représentation de l'image moyenne de l'ensemble des images pour lesquelles celui-ci est sensible. La relation entre cette image moyenne et l'«image synaptique», c'est-à-dire l'image formée par les poids synaptiques du neurone, est donnée par la relation non-linéaire représentée sur la figure 4.23. En outre, il est remarquable que seuls les dispositifs dont les caractéristiques de LTP et de LTD sont modélisables par les équations 4.10 avec un β non nul permettent d'obtenir des poids synaptiques intermédiaires stables pour une large plage de $P(LTP)$, comme le montre la figure 4.25 (voir également (Querlioz et al., 2011b)). Nous avons montré que c'était le cas pour la caractéristique de LTP de la PCM ainsi que pour la nano-synapse de l'Université du Michigan.

Enfin, ces résultats suggèrent que dans cette application précise, un codage semi-fréquentiel est bien adapté. Alors que la fréquence maximale nécessaire est donnée par $1/T_{LTP}$, il n'y a pas de contrainte sur la fréquence minimale. En supposant que les phases initiales des trains d'impulsions sont aléatoires, une intégration des impulsions sur une fenêtre de temps de $1/T_{LTP}$ suffit pour donner une estimation de la valeur moyenne du potentiel de membrane du neurone en régime permanent (en ignorant le terme de fuite). Dans ce cas on peut donc parler de codage semi-fréquentiel, puisqu'il

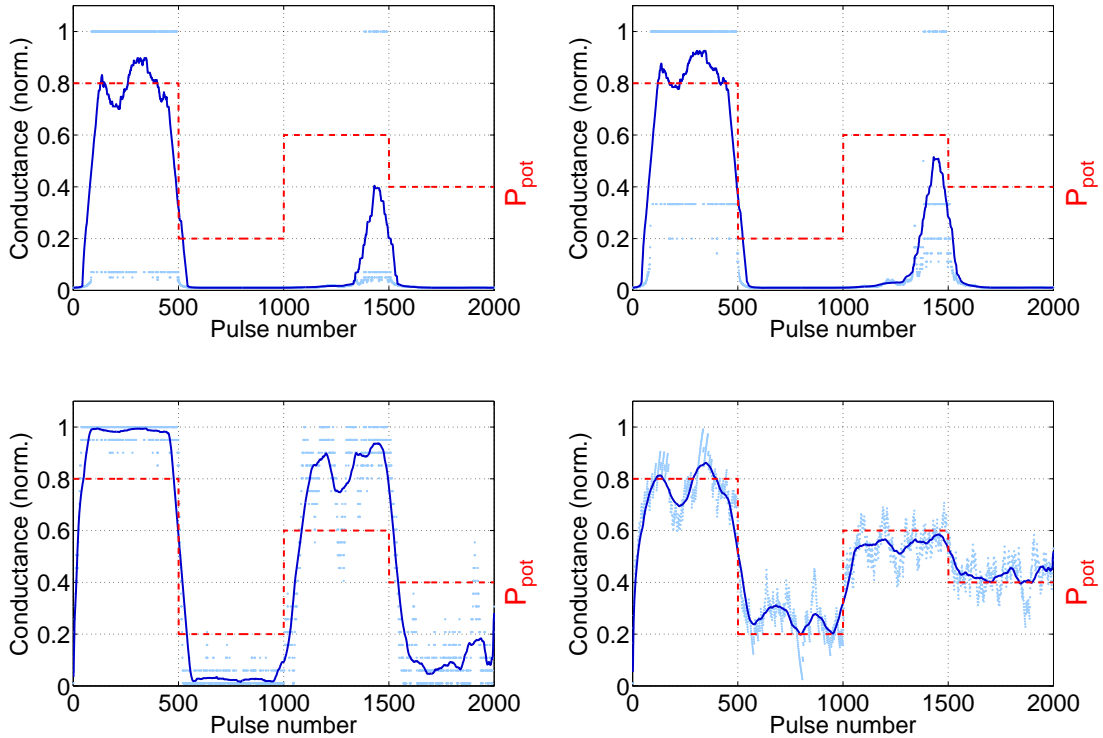


FIGURE 4.25 – Stabilité des poids synaptiques pour différents modèles de synapses en fonction de $P(LTP)$ (P_{pot} sur le graphique). En haut à gauche, le modèle linéaire du memristor de HP Labs. En haut à droite et en bas à gauche, les modèles linéaires par morceaux «classiques», à seuils, sur la résistance et sur la conductance respectivement. Le modèle en bas à gauche est équivalent à un modèle purement additif d'évolution du poids synaptique ($\beta = 0$). En bas à droite, notre modèle décrit aux équations 4.10. Pour le détail des différents modèles, voir (Querlioz et al., 2011b).

n'est pas nécessaire de pouvoir estimer la période des trains d'impulsions de plus basse fréquence, ce qui impliquerait d'en avoir au moins une période, pour que le système fonctionne. On se rapproche donc dans ce cas de l'efficacité d'un codage par rang, en termes de nombres d'évènements. La formalisation présentée ici n'est toutefois pas suffisante pour montrer l'équivalence entre les deux.

Calcul du taux de succès en non supervisé

Le calcul du taux de succès pour l'apprentissage en mode non supervisé repose sur la formation et la stabilisation d'une classe d'équivalence pour chaque neurone. Pour se faire, on stocke en mémoire pour chaque neurone le nombre de fois qu'il a été le plus actif pour chacune des classes des stimuli qui lui ont été présentés. Le neurone le plus actif est celui qui s'est déclenché le plus de fois pour un stimulus donné ou le premier à s'être déclenché si les neurones ne sont autorisés à se déclencher qu'une seule fois par stimulus. Ainsi, comme l'illustre la figure 4.26, on considère que la reconnaissance du stimulus en entrée est bonne si le neurone qui a été le plus actif pour ce stimulus est aussi le plus actif pour la classe d'équivalence à laquelle appartient ce stimulus. Il est ensuite possible de calculer un taux de succès instantané moyen du réseau en faisant une moyenne glissante de **Success** sur les N derniers stimuli présentés.

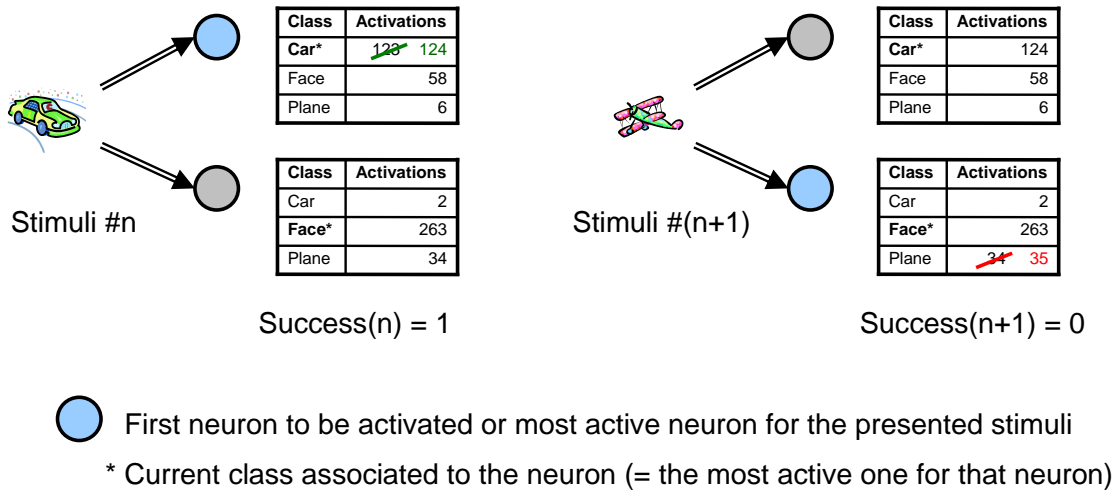


FIGURE 4.26 – Illustration du calcul du taux de succès pour l'apprentissage non supervisé.

4.4.2 Performances comparées avec un Perceptron

Les simulations suivantes ont été effectuées par Damien Querlioz avec l'aide du simulateur Xnet (Querlioz et al., 2011a). L'étude porte sur l'apprentissage et la classification de chiffres écrits à la main. Pour cela, la base MNIST est utilisée, celle-ci consiste en images de 28×28 pixels représentant des chiffres écrits par 250 personnes (Lecun et al., 1998). Cette base est couramment utilisée pour évaluer les performances d'algorithmes d'apprentissage. Pour réaliser un apprentissage, nous présentons l'ensemble de la base MNIST (60 000 chiffres) trois fois au système. Chaque neurone d'entrée est connecté à un pixel de l'image. Il émet une impulsion avec une fréquence proportionnelle à l'intensité du pixel (fréquence maximale de 20 Hz). La phase initiale est aléatoire. Le formalisme introduit précédemment pour STDP est donc applicable. Chaque chiffre est présenté durant 350 ms. Aucun pré-traitement sur les chiffres n'est utilisé et le jeu de données n'est pas augmenté avec des distorsions. Le réseau est ensuite testé avec la base de donnée de test MNIST (10 000 chiffres).

La figure 4.27 représente les poids synaptiques appris par le système dans une configuration avec seulement 10 neurones de sortie. Il est remarquable que sans aucune supervision et en utilisant uniquement notre règle STDP locale, le système a identifié 9 chiffres différents (sur 10), c'est-à-dire les vrais motifs constituant l'entrée. De plus, il a identifié les caractéristiques distinctives de chaque chiffre (et pas simplement le chiffre écrit à la main le plus courant) : la boucle du chiffre deux, le bas du six, ou les parties horizontales consistant les chiffres trois et huit.

Afin d'évaluer quantitativement les performances du réseau, la figure 4.28 montre le taux de reconnaissance durant le processus d'apprentissage pour différents nombres de neurones de sortie. Pour évaluer le taux de reconnaissance, nous associons chaque neurone de sortie au chiffre pour lequel il s'active le plus souvent a posteriori. Matériellement, cette association pourrait être implémentée avec un circuit logique. Cet étiquetage peut également être implémenté avec une deuxième couche supervisée compatible avec la variabilité des dispositifs (Querlioz et al., 2012).

Avec dix neurones de sortie, le taux de reconnaissance atteint 60%. Avec 50 neurones de sortie, celui-ci atteint 81% et avec 300 neurones, 93.5% (avec le même nombre de synapses par neurone). Un réseau de neurones classique avec apprentissage par rétro-propagation du gradient et 300 neurones cachés atteint 95% (Lecun et al., 1998), proche des 93.5% que nous obtenons. Le meilleur algorithme dans la littérature atteint un taux de reconnaissance nettement supérieur de 99.7%, mais avec l'équivalent de 12 millions

de poids ajustables (contre 235 200 ici) et en utilisant un jeu de données d'apprentissage largement augmenté (Cireşan et al., 2010). Bien que nos chiffres soient clairement plus modestes, l'intérêt de notre réseau est qu'il est entièrement non supervisé avec une règle d'apprentissage locale extrêmement simple et non sensible à la variabilité.

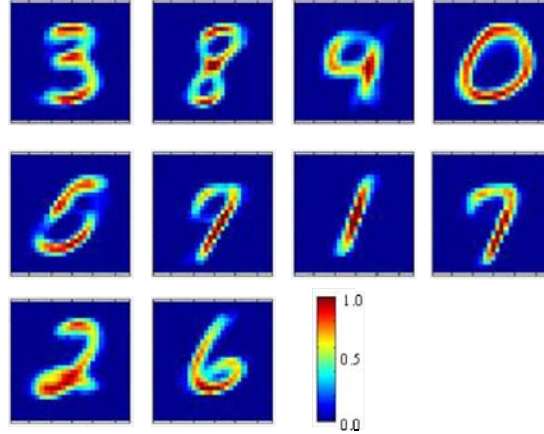


FIGURE 4.27 – Reconstruction des poids appris dans une simulation avec 10 neurones de sortie. Le poids est maximal en rouge et minimal en bleu.

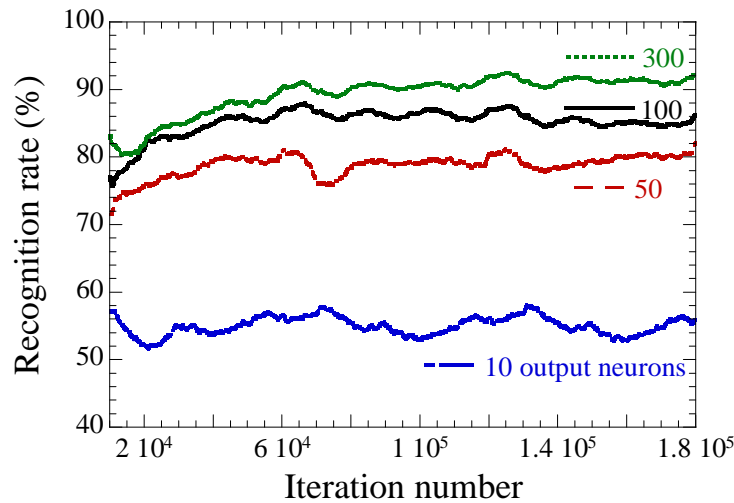


FIGURE 4.28 – Taux de reconnaissance durant l'apprentissage pour différents nombres de neurones de sortie (de bas en haut : 10, 50, 100 et 300). Le taux de reconnaissance est calculé à partir d'une moyenne glissante sur 10 000 itérations.

Robustesse à la variabilité

Les simulations ont montré une tolérance à des niveaux de variabilités impressionnants pour les paramètres synaptiques (Querlioz et al., 2011a). Sont concernés les paramètres modélisant l'évolution des poids synaptiques α_+ , α_- , w_{min} , w_{max} et w_{init} . Une variabilité de 25% sur l'ensemble de ces paramètres n'affecte pas les performances en classification du réseau. Les performances chutent légèrement avec 50% de variabilité. Avec une variabilité extrême de 100%, le système reste fonctionnel et le taux de reconnaissance des chiffres reste de 67% avec seulement 50 neurones. A ce niveau de variabilité, il est notable que 30% des synapses ne peuvent plus être programmées (α_+ ou α_- est nul).

4.5 Discussion et perspectives

Dans ce chapitre, nous espérons avoir pu apporter un premier aperçu convainquant des possibilités et de la richesse de notre approche pour le traitement des données naturelles. Tout d'abord, nous avons introduit le premier système à notre connaissance capable d'extraire de manière exhaustive et non supervisée des motifs se recouvrant temporellement directement à partir de données AER non filtrées, en utilisant une règle STDP simple et locale, ne comportant en tout que 10 paramètres pour les neurones. Nous avons montré que ce type de réseau de neurones impulsionnel peut apprendre à compter des véhicules avec une précision supérieure à 95%, à partir d'une rétine artificielle à la résolution limitée de 128×128 pixels et seulement 10 min de données au temps réel.

Nous avons appliqué ces mêmes principes avec succès sur de l'apprentissage de motifs auditifs, générés à partir d'un modèle de cochlée. Ces travaux proposent en outre un modèle simple capable d'expliquer plusieurs résultats significatifs en psychoacoustique, sur la formation rapide de mémoire auditive robuste. Ce modèle dispose même de qualités prédictives, qui ne manqueront pas d'être explorées dans le futur. Il offre enfin des perspectives intéressantes pour le traitement de données auditives plus complexes, et en premier lieu celui de la parole.

Enfin, nous avons établi des résultats forts sur les capacités de l'apprentissage de STDP pour la classification de motifs, avec un début de formalisme inédit à notre connaissance. La force de ces résultats tient également dans la comparaison avec une application de reconnaissance qui fait référence dans le domaine.

Chapitre 5

Implémentations matérielles sur des technologies memristives

Sommaire

5.1	Introduction	105
5.2	État de l’art	106
5.3	Démonstration d’une mémoire associative sur NOMFET	109
5.3.1	Introduction	109
5.3.2	Caractérisation du NOMFET	110
5.3.3	Méthodologie	111
5.3.4	Résultats expérimentaux	115
5.3.5	STDP asymétrique	119
5.4	Implémentation de STDP sur PCM	120
5.4.1	État de l’art	121
5.4.2	Synapse et architecture proposées	123
5.4.3	Résultats de simulation	128
5.5	Apprentissage STDP stochastique sur CBRAM	133
5.5.1	Synapse et architecture proposées	134
5.5.2	Résultats de simulation	137
5.6	Discussion et perspectives	140

5.1 Introduction

Dans ce chapitre, nous présentons tout d’abord un état de l’art des propositions d’implémentation de «Spike-Timing-Dependent Plasticity» (STDP) sur des dispositifs memristifs. Cet état de l’art, mis en perspective avec nos études d’apprentissage STDP introduites dans le chapitre précédent, nous permet de proposer une implémentation simplifiée adaptée à des dispositifs memristifs dont la conductance peut être modifiée de manière progressive et cumulative. Ceci est le cas pour certains dispositifs expérimentaux de type mémoire résistive – «Resistive RAM» (RRAM), mais des dispositifs plus matures comme la mémoire à changement de phase – «Phase-Change Memory» (PCM) et la «Conductive-Bridging RAM» (CBRAM) n’ont pas cette caractéristique. Pour cette raison, nous proposons dans ce chapitre des implémentations spécifiques pour la PCM et la CBRAM. Dans le cadre de ces travaux, nous avons également réalisé une démonstration de mémoire associative sur le «Nanoparticle-Organic Memory Field Effect Transistor» (NOMFET), qui est un dispositif mémoire volatile, et nous montrons en quoi celle-ci peut être mise en relation avec notre règle d’apprentissage STDP.

Chacune des parties de ce chapitre a fait l'objet d'une ou plusieurs publications et les architectures STDP présentées sur les dispositifs memristifs progressifs, sur la PCM et sur la CBRAM ont fait l'objet d'un dépôt de brevet.

5.2 État de l'art

Les premières propositions d'implémentation d'une règle d'apprentissage STDP sur des dispositifs memristifs remontent à 2008 (Snider, 2008) et 2009 (Linares-Barranco and Serrano-Gotarredona, 2009). Celles-ci concernent des dispositifs memristifs bipolaires dont les changements de conductance sont progressifs et cumulatifs, et qui peuvent donc être modélisés par la caractéristique de la figure 5.1. Cette caractéristique, ou une variante linéaire par morceau, est couramment utilisée pour les modèles de memristor dans les applications neuromorphiques. Elle ne modélise cependant de manière réaliste que quelques dispositifs, de la famille des RRAM, en particulier le memristor de Hewlett-Packard (HP) Labs et la nano-synapse de l'Université du Michigan (ainsi que la PCM dans un quadrant uniquement, pour la cristallisation). Sur ces dispositifs, une faible tension ne modifie pas significativement la conductance et un seuil est généralement défini entre la caractéristique faible champ et celle à tension plus élevée. La règle STDP implémentée émule la forme asymétrique observée en biologie (Bi and Poo, 1998). Ces propositions ne présentent toutefois pas d'application concrète exploitant cette règle d'apprentissage.

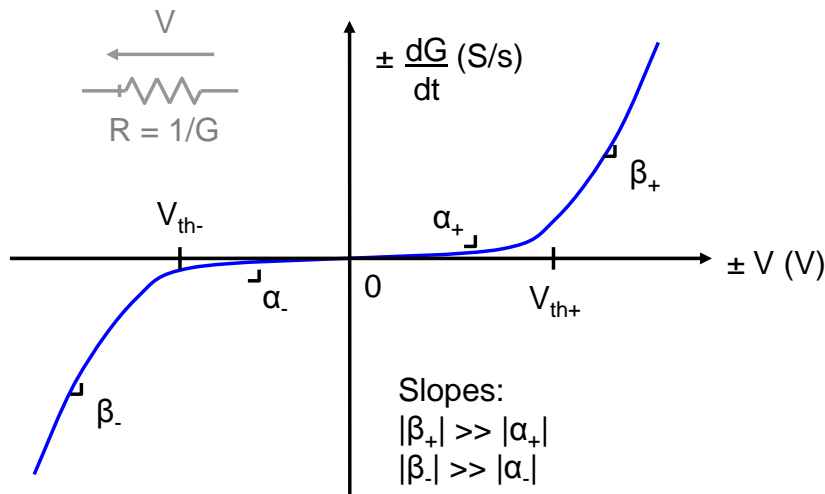


FIGURE 5.1 – Caractéristique dynamique d'un dispositif memristif bipolaire progressif et cumulatif.

Dans (Snider, 2008), la dépendance temporelle de l'amplitude de la potentialisation à long terme – «Long-Term Potentiation» (LTP) ou de la dépression à long terme – «Long-Term Depression» (LTD) avec l'intervalle entre l'impulsion post-synaptique et l'impulsion pré-synaptique est émulée par une série d'impulsions dont la durée diminue dans le temps. Cela est illustré sur la figure 5.2(a). À gauche de la figure, un signal pré-synaptique génère une série d'impulsions dont l'amplitude est insuffisante pour modifier le poids de la synapse. Lorsque le neurone se déclenche plus tard dans le temps, l'impulsion post-synaptique s'ajoute avec l'une des impulsions de la série pré-synaptique. La différence de potentiel résultante dépasse le seuil du dispositif memristif et a pour effet d'augmenter la conductance de la synapse. Si l'on suppose que cette modification est progressive, l'amplitude du changement de conductance sera d'autant plus faible que cette impulsion de programmation est courte, ce qui permet

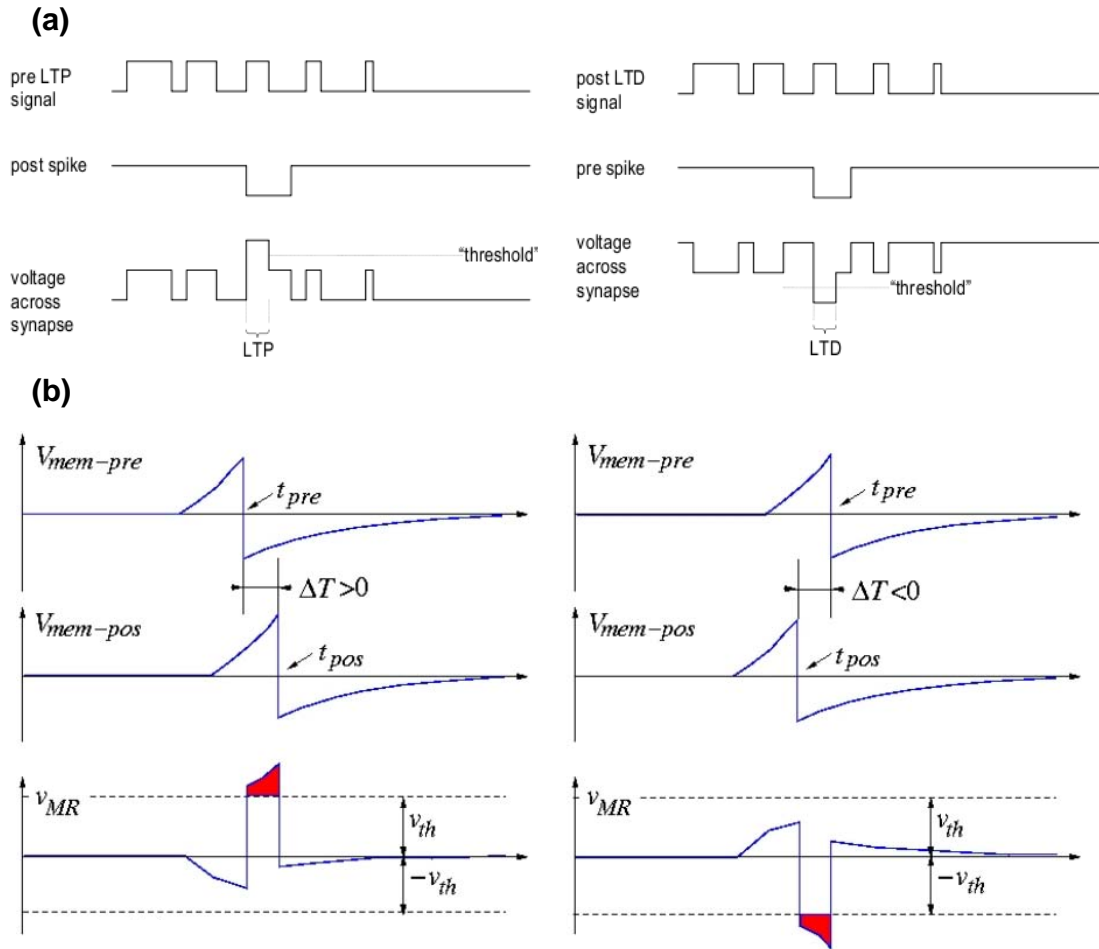


FIGURE 5.2 – Propositions d’implémentation de STDP avec des dispositifs memristifs bipolaires. (a) (Snider, 2008). (b) (Linares-Barranco and Serrano-Gotarredona, 2009).

donc de moduler la LTP avec une série d’impulsions pré-synaptiques de plus en plus courtes. Le principe est identique pour la LTD (à droite de la figure), qui est implémentée avec une série d’impulsions post-synaptiques. Pour que les mécanismes de LTP et de LTD puissent avoir lieu simultanément, un mécanisme de multiplexage temporel «mélange» les signaux de communication, de LTP et de LTD. Dans ce système, une énergie conséquente peut être dépensée dans les très nombreuses charges et décharges des lignes du «crossbar».

Dans (Linares-Barranco and Serrano-Gotarredona, 2009), la modulation de l’amplitude de la LTP/LTD est obtenue par une combinaison entre amplitude et durée des impulsions, comme cela est illustré sur la figure 5.2(b). L’amplitude du changement de conductance est supposée être proportionnelle à l’intégrale de la tension au-dessus du seuil du dispositif. Il est ainsi possible d’obtenir différentes formes de caractéristiques STDP asymétriques en fonction de la forme des impulsions pré- et post-synaptiques. Dans cette proposition d’implémentation, la largeur des impulsions doit être égale aux fenêtres temporelles de la LTP/LTD, qui sont typiquement de l’ordre de 10 à 100 ms pour les applications présentées dans le chapitre précédent. Cela implique que la dynamique du dispositif corresponde à la dynamique de STDP, c’est-à-dire que des impulsions de plusieurs millisecondes ne doivent que modifier légèrement sa conductance. Si le dispositif peut être lu et programmé en quelques microsecondes, cette solution multiplie par mille la consommation énergétique par rapport à un protocole optimal.

Nous avons montré qu’il n’était pas nécessaire d’émuler la forme de STDP biologique

pour mettre au point des applications intéressantes. Au contraire, une forme simplifiée de STDP, avec seulement un niveau de LTP et un niveau de LTD, nous a permis de montrer des applications nouvelles à l'état de l'art. Cette règle STDP est nettement plus simple à implémenter, puisqu'elle n'implique pas de dépendance temporelle entre l'amplitude de la LTP/LTD et les instants pré-post synaptiques. La figure 5.3 présente une implémentation de cette règle avec des dispositifs memristifs bipolaires modélisables par la caractéristique de la figure 5.1.

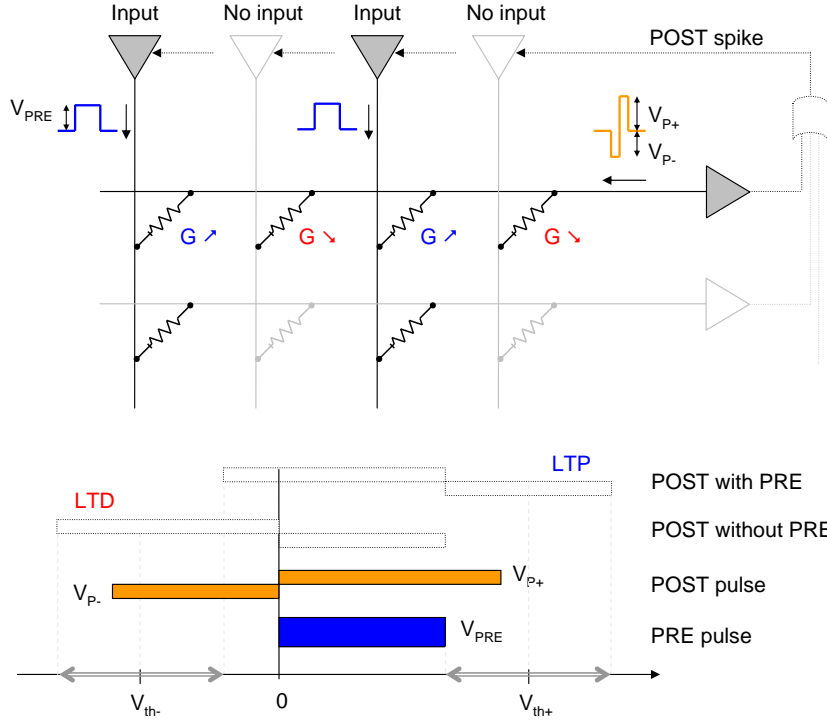


FIGURE 5.3 – Notre proposition d'implémentation de la STDP simplifiée avec des dispositifs memristifs bipolaires.

Le système fonctionne comme suit. Lorsqu'un neurone d'entrée est activé, celui-ci propage une impulsion d'amplitude et de durée minimales vers ses synapses pour permettre l'intégration de leur conductance (poids synaptique) par les neurones de sortie. Le neurone d'entrée passe alors dans un état dit «LTP» pour la durée de la fenêtre de LTP de la règle STDP (implémenté par une machine d'état par exemple). Lorsqu'un neurone de sortie se déclenche, il envoie un signal «POST spike» à l'ensemble de ses neurones d'entrée pour leur signaler une opération de mise à jour des poids synaptiques. Les neurones d'entrée qui sont alors dans un état «LTP» émettent une impulsion de programmation V_{PRE} , qui interagit avec l'impulsion de programmation V_{P+}/V_{P-} émise par le neurone de sortie et induit une LTP sur leur synapse. En cas d'absence d'impulsion V_{PRE} , les synapses subissent une LTD, avec les impulsions telles que représentées en bas de la figure 5.3.

Il est en outre possible de calculer les valeurs optimales des tensions V_{PRE} , V_{P+} et V_{P-} par rapport aux tensions de seuil des dispositifs memristifs V_{th+} et V_{th-} pour autoriser une variabilité maximale sur les seuils (flèches grisées sur l'axe) :

$$V_{PRE} = 2/3.V_{th+} \quad (5.1a)$$

$$V_{P-} = -2/3.V_{th+} \quad (5.1b)$$

$$V_{P+} = 1/3.V_{th+} - V_{th-} \quad (5.1c)$$

5.3 Démonstration d'une mémoire associative sur NOMFET

Ces travaux ont été publiés dans (Bichler et al., 2012c)

5.3.1 Introduction

Dans le concept du conditionnement classique, l'apprentissage associatif implique d'appairer de manière répétitive un stimulus inconditionnel, qui déclenche toujours une action réflexe, avec un stimulus neutre, qui ne provoque normalement pas de réponse. Après conditionnement, une réponse peut être provoquée indifféremment par le stimulus inconditionnel et par le stimulus neutre, qui devient alors un stimulus conditionné. Ce concept remonte aux études d'Ivan Pavlov du début 1900. Dans ses célèbres expériences, il a montré comment un stimulus neutre, comme le son d'une cloche, pouvait être associé à la vue de nourriture et déclencher la salivation de ses chiens (Pavlov, 1927). La mémoire associative est aujourd'hui un concept clé dans l'apprentissage et les capacités d'adaptation du cerveau. Le mécanisme d'association, omniprésent dans la vie de tous les jours, est une conséquence directe de la plasticité du cerveau, qui découle de la plasticité synaptique. Ce concept est à la base de la théorie de Hebb, qui peut être résumée par *les cellules qui s'activent ensemble, se lient ensemble*. L'apprentissage associatif est ainsi largement étudié dans le domaine des réseaux de neurones artificiels (Hassoun, 1993; Norman, 2007).

Cependant, l'absence d'implémentation efficace de synapses artificielles pour des réseaux de neurones à apprentissage associatif a largement limité l'essor de la mémoire associative comme un type de mémoire universel. La nécessité de plusieurs dizaines de transistors pour implémenter une synapse dynamique (Hynna and Boahen, 2006) n'est pas efficace comparée à de la logique reprogrammable. En outre, des milliers de synapses sont généralement nécessaires pour réaliser des fonctions de traitement sur des données réelles, comme le traitement de stimuli visuels ou auditifs par exemple (Bichler et al., 2012a), avec une possibilité de réutilisation limitée pour d'autres tâches. Des synapses implémentées avec la technologie «Complementary Metal Oxide Semiconductor» (CMOS) la plus moderne seraient encore éloignées de plusieurs ordres de grandeur, en termes de surface et de consommation énergétique, à des synapses biologiques. Celles-ci atteignent une densité de $\sim 10^{12}$ synapses/cm³ et une consommation de 1-10 fJ/potentiel d'action si l'on considère un taux de décharge moyen de 1-100 Hz et une consommation énergétique du cerveau humain de l'ordre de 20 W (Kandel et al., 2000).

Outre les technologies déjà évoquées pour implémenter des synapses artificielles, les dispositifs synaptiques volatiles et/ou organiques ont reçu relativement peu d'attention de la part de la communauté. De la mémoire organique bi-stable basée sur des nanocomposés polyméthacrylate de méthyle – «Polymethyl Methacrylate» (PMMA) :ZnO (Ramana et al., 2012) ou PMMA :C60 (Frolet et al., 2012) a été démontrée récemment. De la mémoire multi-niveaux et non volatile a également été démontrée, par exemple avec des transistors organiques à nanofils de silicium, en contrôlant la concentration d'ions dans la couche mince via la tension de grille (Lai et al., 2008). Bien que les processus physiques modulant la conductance pourraient bien être cumulatifs, la faisabilité d'une implémentation de LTP ou de LTD par changements graduels de conductance reste à évaluer. Cela nous conduit au NOMFET (Alibart et al., 2010) (Bichler et al., 2010). Les précédents travaux sur ce dispositif (Alibart et al., 2012) ont montré que le NOMFET peut être vu comme un dispositif memristif, en modulant la conductivité du canal via la charge des nano-particules déposées dans l'intervalle inter-électrodes composé d'un semi-conducteur organique. Il est volatile et a un temps de rétention compris

entre 10 et 1000 s. Avec ces propriétés physiques, le NOMFET possède de nombreuses caractéristiques d'une synapse biologique (Abbott et al., 1997; Tsodyks et al., 1998), quand il est utilisé en régime impulsionnel, avec un effet cumulatif clair.

Une mémoire associative élémentaire constituée de dispositifs memristifs a été proposée pour la première fois par Yuriy Pershin et Massimiliano Di Ventra, reproduisant l'expérience du chien de Pavlov (Pershin and Di Ventra, 2010a). Pour leur démonstration, les dispositifs memristifs ont toutefois été émulés à l'aide d'un microcontrôleur, éliminant ainsi toutes les difficultés liées à un interfaçage avec de vrais nano-dispositifs physiques, qui ne peuvent pas être prévues en simulation avec un simple modèle fonctionnel. Une autre approche a été proposée par (Ziegler et al., 2012) pour réaliser la même mémoire associative élémentaire, en utilisant un unique dispositif basé sur une structure Pt/Ge_{0.3}Se_{0.7}/SiO₂/Cu. Dans ce sous-chapitre, nous proposons un système original basé sur des NOMFETs pour implémenter une mémoire associative dynamique et le démontrons en interfaçant les NOMFETs avec un circuit CMOS discret. Nous montrons comment les propriétés synaptiques uniques du NOMFET peuvent être utilisées pour implémenter ce que nous désignons par une mémoire associative *dynamique*, pour laquelle l'association est conservée uniquement tant qu'une activité minimale est maintenue en entrée. Dans la dernière partie, la dynamique d'apprentissage est mise en relation avec la volatilité du NOMFET, le mécanisme d'apprentissage est interprété en termes de STDP et l'impact de la variabilité est brièvement discuté.

5.3.2 Caractérisation du NOMFET

La figure 5.4 montre la caractéristique dynamique de charge/décharge du NOMFET. Le courant est mesuré avant et après qu'une impulsion de programmation de 10 s soit appliquée sur V_G . La caractéristique $\Delta I/I - V$ montre comment la conductivité est modifiée en fonction de la tension de l'impulsion, avec une non-linéarité entre 0 et 15 V. Cette caractéristique reste valide lorsque le courant transitant dans le canal est nul, ce qui fait du NOMFET un dispositif memristif idéal (Alibart et al., 2012) dans le sens où en moyenne, aucune puissance n'est dissipée pour la programmation de sa conductivité. L'évolution de la conductivité est contrôlée par l'intégrale temporelle de la tension de grille, contrairement au memistor (à ne pas confondre avec le memristor), qui était programmé via l'intégrale temporelle du courant entrant dans sa troisième électrode (Widrow et al., 1961). Celui-ci utilisait la déposition d'un électrolyte liquide sur un substrat résistif pour moduler sa résistance et n'était par conséquent pas intégrable, contrairement aux dispositifs à électrolytes solides tels que la CBRAM.

Bien que nous ayons rapporté un comportement synaptique pour des NOMFETs courts (intervalle inter-électrodes de 200 nm et nano-particules (NPs) de 5 nm), fonctionnant à des tensions de polarisation de -3 V (Alibart et al., 2010), nous avons utilisé pour cette étude des NOMFETs plus longs (intervalle inter-électrodes de 5 μ m et NPs de 20 nm), parce qu'ils possèdent une plasticité d'amplitude plus grande (c'est-à-dire une modulation plus importante de la conductivité sous l'action d'impulsions de programmation). Avec un courant à l'état passant (ON) entre 50 et 500 nA et une tension drain-source de 15 V, la résistance équivalente de la synapse se situe entre 30 M Ω et 300 M Ω . Bien que ces valeurs soient relativement élevées comparées à d'autres technologies memristives (Bichler et al., 2012b) (Kund et al., 2005), cela se fait sans sacrifier le rapport OFF/ON grâce à l'effet de champs du transistor (courant à l'état bloqué (OFF) inférieur à 0.1 nA, précision limitée par les instruments de mesure). Cela fait du NOMFET un dispositif faible consommation et malgré tout utilisable dans des matrices de dispositifs pour des réseaux à grande échelle (Liang and Wong, 2010).

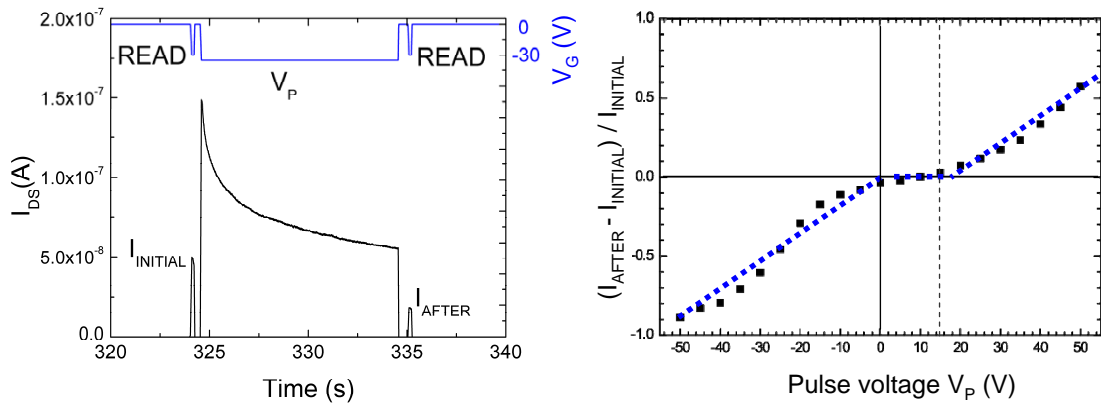


FIGURE 5.4 – Caractéristique dynamique du NOMFET : changement relatif de conductivité en fonction de la tension des impulsions. Le courant est mesuré juste avant et après l'application d'une impulsion de 10 s avec différentes amplitudes. Le changement relatif du courant est lié à la quantité de charge dans les NPs. Pour des impulsions négatives, le NOMFET est en accumulation, les NPs sont chargées positivement par les trous, ce qui réduit la conductivité par effet de Coulomb. Pour des polarisations positives, les trous sont dé-piégés, conduisant à une augmentation du courant. La courbe est non-linéaire avec une zone de 0 V à 15 V où la charge des NPs n'est pas affectée par la tension appliquée. Pour plus de détails, voir (Alibart et al., 2012).

5.3.3 Méthodologie

Notre mémoire associative est constituée de deux neurones d'entrée, deux synapses et un neurone de sortie (figure 5.5). Une synapse est implémentée avec un unique NOMFET et les neurones sont construits avec des composants CMOS discrets sur une carte électronique. L'association est réalisée exclusivement grâce à la plasticité du NOMFET : sa conductivité est modifiée de manière non supervisée par l'interaction d'impulsions pré-synaptiques (signal d'entrée) et d'impulsions post-synaptiques (rétroaction), de manière à ce que l'association se produise lorsqu'il y a une corrélation temporelle entre les impulsions pré- et post-synaptiques sur une durée suffisamment longue (voir le lien avec STDP dans la partie «Résultats expérimentaux»).

Les opérations sur les synapses sont faites en deux étapes, une étape de lecture et une étape d'écriture. Le fait de séparer ces deux étapes autorise plus de contrôle sur la dynamique du système, en découplant la plasticité physique des NOMFETs de la dynamique d'apprentissage. De manière plus générale, même pour des réseaux de neurones à base de dispositifs memristifs non volatiles, séparer ces deux étapes peut avoir plusieurs avantages : parce que la lecture peut être faite avec une impulsion courte et/ou de faible tension, le changement de la conductivité du dispositif causé par la lecture peut être minimisé, tout comme la consommation énergétique. Un autre intérêt de cette approche est qu'aucun courant n'a besoin de circuler dans le dispositif durant l'étape d'écriture. En effet, contrairement aux dispositifs memristifs basés sur la résistance, les nano-particules dans le canal du NOMFET peuvent être chargées ou déchargées sans que le transistor ne soit dans un état passant, modulant ainsi la conductivité du canal sans courant drain-source. Nous utilisons ce principe dans notre système expérimental, où les impulsions pré-synaptiques sont appliquées sur la grille des NOMFETs. Le neurone de sortie est connecté aux terminaux de source et drain de tous les dispositifs et peut appliquer des tensions différentes à la source et au drain.

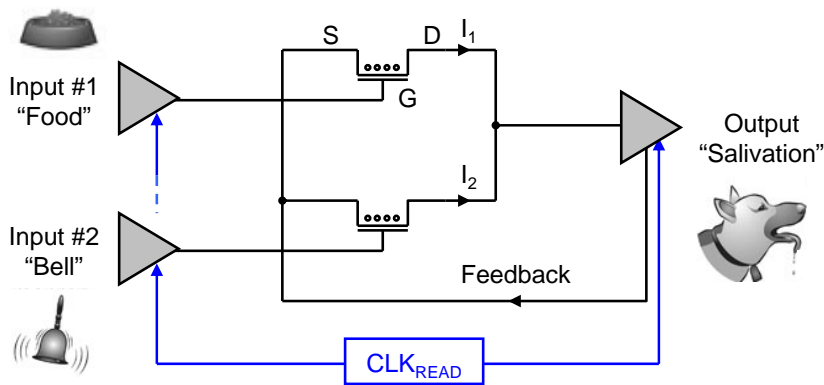


FIGURE 5.5 – Circuit électronique équivalent pour la mémoire associative. Il y a 3 neurones (Entrée #1, Entrée #2 et Sortie) et 2 synapses/NOMFETs. Un signal d’horloge global (CLK_{READ}) permet la synchronisation des neurones d’entrée et de sortie durant les opérations de lecture. I_1 et I_2 sont les courants drain-source mesurés.

Lecture et activation du neurone de sortie

Une étape de lecture a lieu périodiquement et déclenche l’activation du neurone de sortie si une entrée est présente sur une synapse fortement conductrice (signifiant que le neurone «écoute» la synapse). Durant cette étape, l’état de conduction des dispositifs est mesuré. Une rétroaction est générée si un signal d’entrée actif est présent sur l’un des NOMFETs durant la lecture et que sa conductivité est supérieure à un seuil prédéterminé. L’étape de lecture implique que les neurones d’entrée et de sortie soient synchronisés, car les neurones d’entrée actifs doivent rendre les NOMFETs passants en appliquant une tension négative à leur grille. Le neurone de sortie doit appliquer une impulsion négative sur la source des dispositifs et mesurer le courant drain-source sur l’électrode de drain. Ce processus peut être effectué de manière séquentielle, en ne rendant passant qu’un seul NOMFET à la fois, de manière à ce que tous les terminaux de drain et tous les terminaux de source puissent être connectés ensemble.

Écriture (interaction pré-post)

La conductivité des NOMFETs est modulée lors des étapes d’écriture, via l’interaction des impulsions pré-synaptiques (sur la grille) et des impulsions post-synaptiques (sur la source et le drain). Pour réaliser une mémoire associative, la conductivité d’un NOMFET ne doit augmenter que dans le cas d’impulsions simultanées pré- et post-synaptiques, signifiant qu’il y a une corrélation entre les événements menant à l’activation du neurone de sortie et les événements entrant sur la synapse. Cela est implémenté en assurant que seule une interaction des impulsions pré- et post-synaptiques puisse produire une augmentation significative de la conductivité du NOMFET, en appliquant une tension équivalente entre la grille et les terminaux drain-source supérieure à celle obtenue avec une impulsion pré- ou post-synaptique seule. Comme la relation entre le changement de conductivité et l’amplitude de la tension appliquée n’est pas linéaire (voir figure 5.4), il est possible de maximiser l’effet de l’interaction de deux impulsions et de minimiser l’effet d’une seule impulsion sur la conductivité. La forme des impulsions est donnée figure 5.6 :

- Lorsqu’aucune rétroaction n’est présente, l’entrée n’a qu’un effet limité sur la conductivité de la synapse : une absence de signal d’entrée signifie une tension constante de -15 V sur la grille, qui a tendance à diminuer la conductivité au cours du temps (perte de mémoire). Cela est compensé lorsque l’entrée est active, où la conductivité augmente légèrement. Cela a globalement pour effet de maintenir

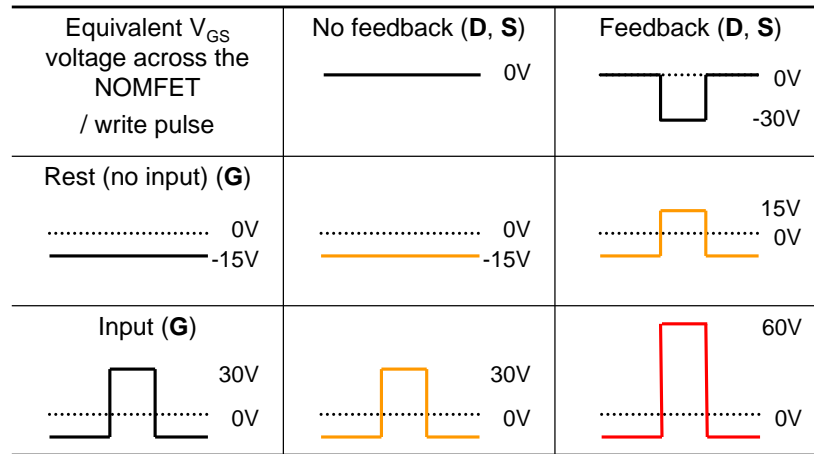


FIGURE 5.6 – Forme des impulsions d'entrée et de rétroaction pour implémenter l'apprentissage associatif. Les signaux d'entrée sont appliqués à la grille et la rétroaction aux terminaux de drain et de source. Les entrées sont polarisées négativement, ce qui signifie qu'une tension de -15 V est appliquée en permanence sur les entrées en l'absence de signal. Seule l'interaction d'un signal d'entrée et de rétroaction crée une tension de programmation entre la grille et les terminaux drain-source suffisamment importante pour augmenter significativement la conductivité du NOMFET, produisant ainsi une association lorsque plusieurs interactions ont lieu durant une courte durée.

la conductivité de la synapse tant qu'une activité est maintenue à son entrée.

- Lorsqu'il y a une rétroaction, il n'y a pas de changement de conductivité en l'absence d'entrée, tandis qu'une augmentation majeure de la conductivité se produit lorsqu'un signal d'entrée est présent en même temps que l'impulsion de rétroaction. La polarisation de -15 V a été choisie pour éviter que l'association ne se maintienne uniquement grâce à la rétroaction (la tension aux bornes du NOMFET est donc de 15 V avec rétroaction, au lieu de 30 V avec l'entrée).

Pendant les opérations d'écriture, les électrodes de source et de drain sont connectées, ce qui implique que la seule différence de potentiel dans le dispositif se situe au niveau de la grille, quelque soit la simultanéité des impulsions d'entrée et de rétroaction. Dans cette configuration, le dispositif se comporte comme un condensateur, avec la charge et la décharge des NPs modulant la conductivité du transistor pour une tension de grille donnée lors de la lecture. Comme dans un condensateur, aucun courant ne peut traverser la grille, et le changement de conductance ne dissipe pas de puissance dans le dispositif. Cela contrairement aux dispositifs mémoire résistifs, non basés sur la charge et non volatiles, comme la PCM ou la CBRAM, pour lesquels une modification de la conductance nécessite de l'effet Joule pour obtenir un changement de phase ou une migration d'ions dans le matériau. Dans ces technologies, le courant de programmation se situe typiquement dans le domaine du mA ou du μA (Bichler et al., 2012b) (Kund et al., 2005).

Calibration

Dans un réseau d'apprentissage associatif dynamique, une étape de calibration est nécessaire pour initialiser l'état des synapses, afin de programmer le système pour répondre à un jeu initial de stimuli inconditionnels. Un seuil optimal peut aussi être automatiquement calculé pour cette configuration initiale. Comme les poids synaptiques sont programmés en parallèle, cette configuration initiale ne dépend pas du nombre de synapses.

Dans notre démonstrateur, ce processus de calibration est automatisé dans le micro-

contrôleur implémentant les neurones (voir les sections suivantes). Ce processus fixe le seuil avant le début de l'apprentissage associatif. En plus, un gain constant est automatiquement appliqué à la lecture de l'un des deux NOMFETs pour compenser la variabilité entre les deux dispositifs. La séquence de calibration est la suivante : (1) Mesure de la conductivité de chaque dispositif pour les quatre états de programmation possibles (avec ou sans impulsion pré-synaptique et avec ou sans impulsion post-synaptique). (2) Calcul du gain pour le second dispositif pour compenser la dispersion. (3) Calcul du seuil optimal. (4) Initialisation de la conductivité des deux dispositifs.

Il est clair que les étapes (1)-(2) seraient certainement difficilement réalisables dans un système plus complexe. Cependant, la compensation de la variabilité sur la caractéristique statique I_{DS} pourrait être éliminée pour des dispositifs assortis sur la même puce, fabriqués dans un processus industriel contrôlé. En effet, nous avons constaté que la plus grande part de la variabilité était causée par la connexion et l'encapsulation des dispositifs.

Carte électronique expérimentale

Nous avons réalisé une carte électronique spécifique pour interfacer les NOMFETs. Un ingénieur de l'Institut d'Électronique, de Microélectronique et de Nanotechnologie (IEMN) a encapsulé les NOMFETs dans un boîtier métallique standard traversant de type «Transistor Outline» (TO). La conception de cartes électroniques avec des composants discrets standards pour interfacer des dispositifs expérimentaux est souvent difficile, sans mentionner les contraintes spécifiques sur la température ou le contrôle de l'atmosphère pour la connexion («wire bonding») et l'encapsulation selon les matériaux utilisés. Pour le NOMFET, il faut être capable de mesurer des courants de l'ordre du nanoampère avec des tensions allant jusqu'à ± 30 V pour la première génération du composant, utilisée dans ce système.

Malgré ces possibles difficultés, les NOMFETs présentés fonctionnent dans une atmosphère ambiante sans encapsulation spécifique et sont restés stables sur une durée de plus de 6 mois. Nous avons déjà montré précédemment que les propriétés synaptiques du NOMFET sont maintenues avec une épaisseur de grille de 10 nm, un intervalle inter-électrodes de 200 nm et des NPs de diamètre 5 nm (Alibart et al., 2010), réduisant ainsi la tension de programmation maximale à 3 V. L'approche «bottom-up» et la faible température de déposition pour la fabrication des NOMFETs est compatible avec la technologie CMOS et nous pensons que l'interfaçage de dispositifs expérimentaux avec de l'électronique standard est un pas important pour la réalisation de prototypes à plus grande échelle et la co-intégration avec le CMOS.

Comme la mémoire associative proposée ne demande que quatre niveaux de tension différents pour générer les impulsions de lecture et d'écriture, les impulsions à chaque électrode sont générées avec un multiplexeur analogique, le MAX14752, qui peut commuter des tensions jusqu'à ± 36 V. Puisque nous utilisons la dynamique du NOMFET sur des plages de la milliseconde ou de la seconde, la vitesse de commutation n'est pas critique.

Pour l'étape de lecture, la conductivité est obtenue en mesurant le courant passant dans l'électrode de drain lorsqu'une impulsion positive est appliquée à la source, ce qui est équivalent à appliquer une tension négative sur la grille et le drain. La mesure du courant est réalisée avec un convertisseur courant-tension utilisant une topologie d'ampèremètre à rétroaction («feedback-ammeter»). Dans cette topologie, l'amplificateur opérationnel délivre une tension proportionnelle au courant circulant dans la résistance de rétroaction, qui est égal au courant drain-source du NOMFET. Cette topologie est particulièrement adaptée pour la mesure de courants faibles (inférieurs au μA). Un OPA445 est utilisé pour cette fonction, c'est l'un des rares amplificateurs opérationnels

qui supporte une tension allant jusqu'à ± 45 V et qui a un offset en courant d'entrée faible (de l'ordre de 10 pA), requis pour mesurer les courant jusqu'au nA. Il est suivi d'un amplificateur différentiel AD629B pour permettre la mesure de courant pendant qu'une impulsion de rétroaction est appliquée au NOMFET, en supprimant le mode commun introduit par celle-ci. Il supporte une tension de mode commun élevé et a un faible offset de tension d'entrée. Enfin, un convertisseur analogique numérique (CAN) LTC1856 permet d'obtenir une valeur numérique du courant. Il peut mesurer des tensions bipolaires allant jusqu'à ± 10 V directement sur ses entrées et peut supporter des tensions de ± 30 V sans dommage. Il possède également huit canaux multiplexés avec une résolution de 16 bits, ce qui rend facile la mesure de plusieurs dispositifs avec un seul composant.

Une photographie de la carte est montrée figure 5.7 et un schéma simplifié du circuit figure 5.8. Les précautions particulières pour mesurer des courants à l'échelle du nA ont été un routage soigneux, incluant un plan de masse et l'utilisation de capacités de découplages adéquates.

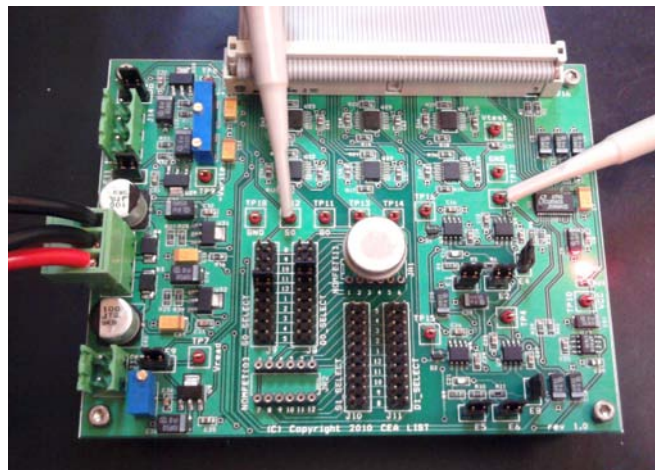


FIGURE 5.7 – Photographie de la carte expérimentale, avec les NOMFETs encapsulés dans un boîtier TO au centre. L'interface avec l'ordinateur, incluant la carte de contrôle FPGA implémentant la logique de contrôle pour les neurones, n'est pas montrée.

Émulation des neurones

Comme la carte expérimentale fournit une interface numérique-analogique (génération des impulsions) et analogique-numérique (mesure des courants) avec les NOMFETs, les neurones sont implémentés numériquement sur une carte basée sur un FPGA, qui gère la calibration initiale et supervise le processus d'apprentissage. Plutôt que d'implémenter la logique de contrôle directement avec une machine d'état en langage de description matériel, une approche programmatique a été préférée pour plus de flexibilité. Pour se faire, un cœur de micro-contrôleur Microblaze de Xilinx a été synthétisé sur le FPGA et programmé en C.

5.3.4 Résultats expérimentaux

Afin de démontrer le fonctionnement de notre mémoire associative, nous avons reproduit l'expérience du chien de Pavlov. La figure 5.9 montre le processus d'association : les deux entrées du système modélisent respectivement la vue de nourriture et l'audition du son d'une cloche, tandis que le neurone de sortie modélise l'activité de salivation du chien. Au début, la conductivité des dispositifs est programmée de ma-

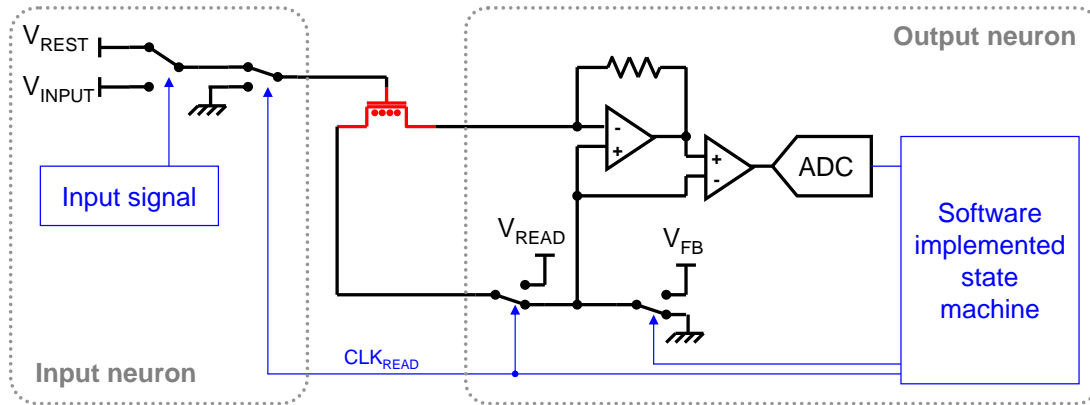


FIGURE 5.8 – Schéma simplifié de la carte expérimentale pour un NOMFET. Les parties bleues/-fines sont implémentées logiquement sur le Microblaze synthétisé sur le FPGA. Dans le mode de lecture, l'électrode de source est connectée à V_{READ} , tandis que les deux autres terminaux sont mis à la masse. Lorsque le neurone de sortie s'active, les terminaux de source et de drain sont connectés à V_{FB} et la conductivité du dispositif est modulée en fonction de la tension de grille, qui peut valoir V_{REST} (pas d'entrée) ou V_{INPUT} (entrée).

nière à ce que seule la vue de nourriture (stimulus inconditionné) produise l'activation du neurone de sortie (salivation). Avant que l'association n'ait été faite, l'audition de la cloche (stimulus neutre) ne déclenche pas de salivation, car la conductivité de la synapse correspondante est sous le seuil du neurone. Lorsque les deux entrées sont actives simultanément, la conductivité de la seconde synapse augmente jusqu'à atteindre le seuil (conditionnement) : l'association est faite. A partir de ce moment, le son de la cloche seul suffit à déclencher le neurone de sortie (stimulus conditionné). Sur la figure 5.11, on peut vérifier que sans rétroaction du neurone de sortie, le changement de conductivité induit uniquement par les impulsions pré-synaptiques n'est pas suffisant pour créer une association.

Notre circuit d'apprentissage associatif est symétrique, ce qui signifie qu'il n'y a aucune différence entre l'entrée recevant le stimulus inconditionné (vue de nourriture) et l'entrée recevant le stimulus neutre (son d'une cloche). L'entrée inconditionnée est celle initialement programmée pour avoir un poids synaptique supérieur au seuil du neurone de sortie. Ce principe est différent du système asymétrique introduit par (Ziegler et al., 2012), dans lequel les deux entrées ne sont pas interchangeables.

Volatilité et dynamique d'apprentissage

Comme la conductivité du NOMFET est volatile, ce que nous implémentons réellement est une mémoire associative à court terme. En effet, lorsqu'il n'y a pas d'activité soutenue aux entrées, l'association est perdue (figure 5.10). Cette fonctionnalité est directement fournie par la physique du dispositif. Il n'y a pas besoin d'un circuit additionnel qui viendrait périodiquement diminuer la conductivité des synapses. Cette fonctionnalité est en effet essentielle pour toute implémentation pratique de mémoire associative, car sinon, les poids synaptiques du système finiraient par saturer à cause des courtes corrélations aléatoires entre les entrées.

Le NOMFET présente à la fois une dynamique court terme et une dynamique long terme, qui sont facilement identifiables sur la figure 5.9. La dynamique court terme correspond aux augmentations et diminutions rapides de la conductance, respectivement lors des activations et désactivations des neurones d'entrée (dynamique temporelle typique ~ 1 s). L'évolution moyenne du courant, ou de la conductance, durant l'apprentissage, suit une dynamique à plus long terme (dynamique temporelle typique

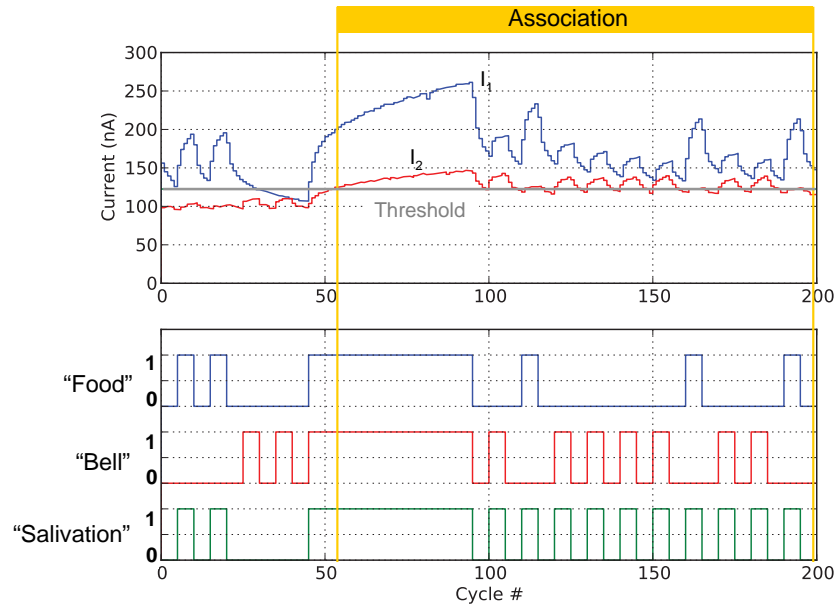


FIGURE 5.9 – Dynamique de l'apprentissage associatif : l'association est maintenue tant qu'il y a une activité minimale sur les entrées. La durée d'un cycle est de 200 ms. Après chaque cycle, le courant est mesuré et comparé au seuil du neurone. Dans cette expérience, la durée minimale d'une impulsion d'entrée est de 5 cycles (1 s). La durée totale de la séquence est de 40 s.

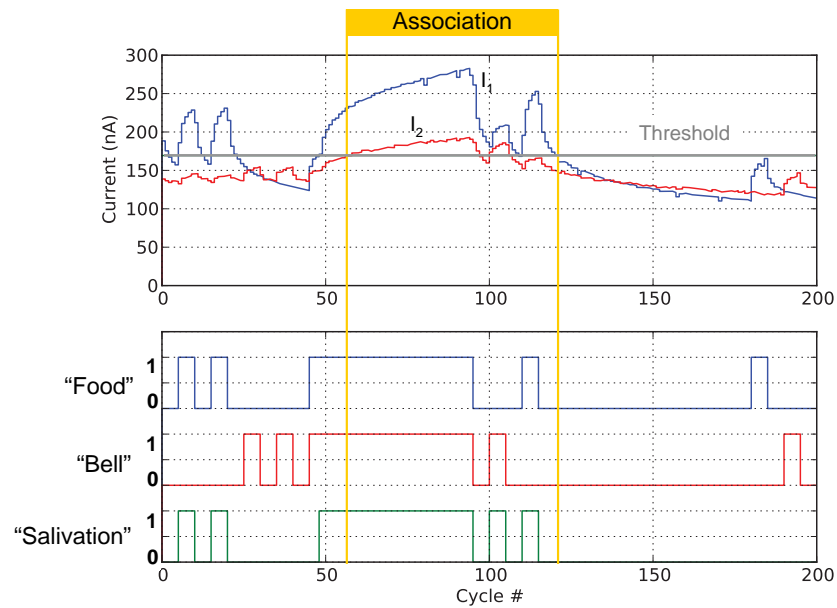


FIGURE 5.10 – Apprentissage associatif dynamique : si l'activité aux entrées n'est pas maintenue, l'association est oubliée.

~ 10 à 100 s) : le courant moyen est en effet supérieur après l'association qu'avant. Une impulsion de potentialisation longue (étape de conditionnement) est nécessaire pour induire une modification durable du poids synaptique. Une telle dynamique synaptique serait probablement difficile à émuler avec des dispositifs memristifs non volatils, comme la PCM, car chaque dispositif nécessiterait une référence de temps externe pour implémenter sa dynamique temporelle.

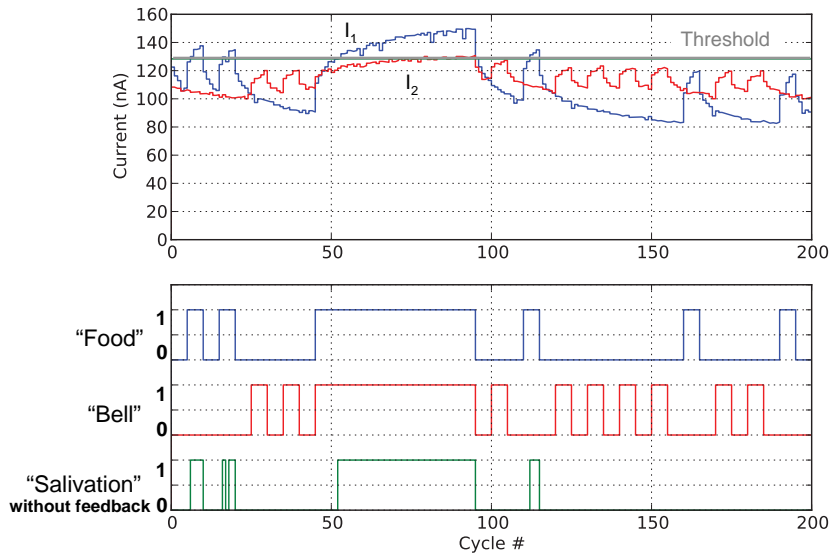


FIGURE 5.11 – Même séquence, sans rétroaction : aucune association n’est faite et la première synapse perd également son association. Comme dans les autres expériences, le meilleur seuil est automatiquement calculé au début de la séquence en utilisant la même procédure de calibration, dans laquelle la rétroaction a également été désactivée.

Relation avec STDP

STDP est actuellement l’un des algorithmes d’apprentissage neuromorphique les plus étudiés sur les dispositifs memristifs (Yu and Wong, 2010; Kuzum et al., 2012; Jo et al., 2010) (Bichler et al., 2012b). Il n’y a pas de règle STDP unique : une large famille de caractéristiques d’évolution du poids synaptique en fonction de la différence temporelle entre les impulsions pré-post a été rapportée (Wittenberg and Wang, 2006). Plus précisément, de la STDP asymétrique a été observée avec les synapses glutamatergiques de l’hippocampe (Bi and Poo, 1998) et de la STDP symétrique a été observée dans la région CA1 de l’hippocampe (Nishiyama et al., 2000) et dans les synapses GABAergiques (Woodin et al., 2003).

La STDP asymétrique a été rapportée avec le NOMFET dans (Alibart et al., 2012). Le procédé de programmation réalisé avec notre mémoire associative émule au contraire une forme de règle d’apprentissage de type STDP symétrique, comme illustré qualitativement sur la figure 5.12. En effet, une augmentation significative de la conductivité est obtenue uniquement lorsqu’il y a simultanément des événements pré- et post-synaptiques (impulsions d’entrée et de rétroaction), produisant une tension équivalente de 60 V aux bornes du NOMFET.

Impact de la variabilité

Bien qu’il y ait un facteur 10 sur la conductivité moyenne entre les deux NOMFETs utilisés dans cette expérience (avant le gain correcteur), la variabilité sur le comportement dynamique des dispositifs est très faible en comparaison (Bichler et al., 2010; Alibart et al., 2012). Comme notre mémoire associative est basée sur un codage temporel, seule la variation relative de la conductivité obtenue avec les impulsions de programmation et la relaxation naturelle des NPs imposent la dynamique de l’apprentissage. A cet égard, il a déjà été montré que les systèmes basés sur un apprentissage STDP étaient tolérants à des variations sur les pas de mise à jour des poids synaptiques allant de 20% à 100% (Querlioz et al., 2011a) (Bichler et al., 2012a). En conséquence, les variations de courant des deux NOMFETs ne sont pas forcément identiques suite au processus

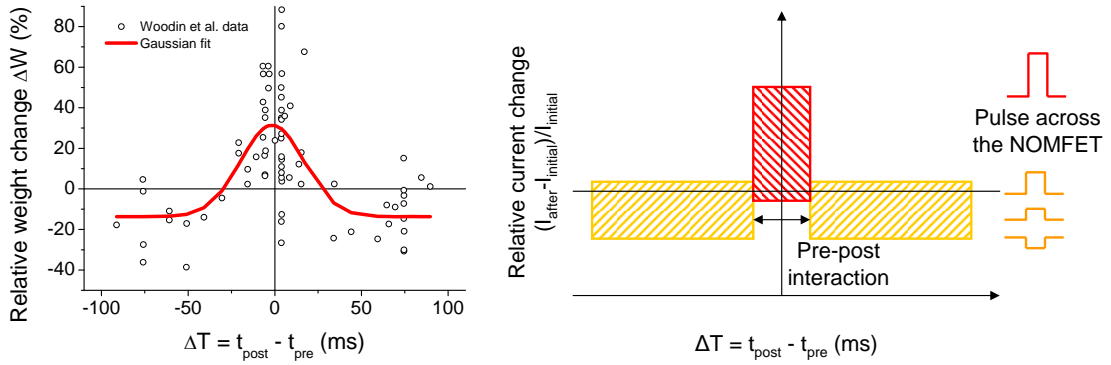


FIGURE 5.12 – Gauche : modification relative du poids synaptique dans la STDP symétrique, observée dans les synapses GABAergiques. Données expérimentales issues de (Woodin et al., 2003). Droite : modification relative de la conductance du NOMFET en fonction des impulsions pré-post pour notre apprentissage associatif (diagramme qualitatif).

de calibration, comme cela est apparent sur les figures 5.9-5.10. Et pour cause, l'effet du processus de calibration est de faire correspondre la dynamique temporelle des variations de conductivité. En outre, cette dynamique reste valide pour des échelles de temps plus longues, avec des durées de cycle plus élevées d'un ordre de grandeur.

5.3.5 STDP asymétrique

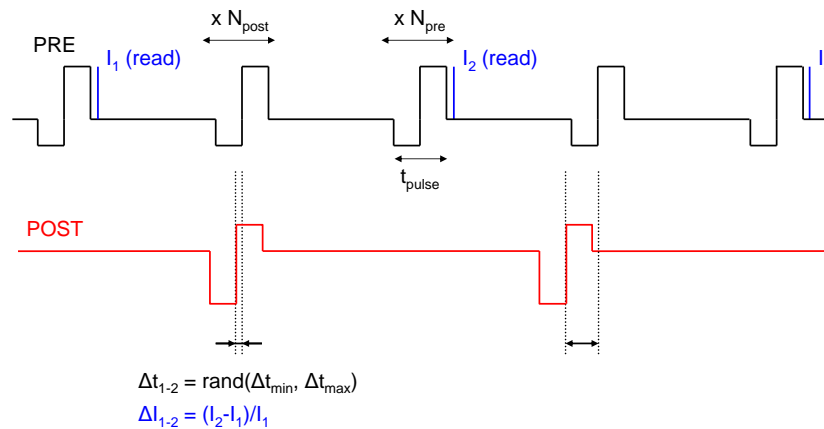


FIGURE 5.13 – Forme des impulsions appliquées à l'entrée (impulsions pré-synaptiques) et à la sortie (impulsions post-synaptiques) du NOMFET pour la mesure des données de la figure 5.14. Pour chaque Δt , une première séquence de $N_{\text{pre}} = 20$ impulsions pré-synaptiques est appliquée avant que la conductivité ne soit mesurée, suivie d'une séquence de $N_{\text{post}} = 20$ interactions d'impulsions pré-post, après laquelle la conductivité est à nouveau mesurée et comparée à la précédente pour obtenir le changement de conductivité. La durée des impulsions pré-synaptiques est de 4 s (-15 V durant 2 s suivi de 30 V durant 2 s), comme celle des impulsions post-synaptiques (-30 V durant 2 s suivi de 15 durant 2 s).

Pour démontrer qu'il est également possible de réaliser une STDP asymétrique avec notre démonstrateur, une série d'impulsions rectangulaires sont appliquées simultanément sur deux NOMFETs, avec un intervalle Δt aléatoire entre les impulsions pré- et post-synaptiques. La variation relative de courant $\Delta I / I$ mesurée simultanément pour les deux NOMFET en fonction de Δt est reportée sur la figure 5.14. La fonction STDP obtenue avec ce système NOMFET/CMOS est en accord avec celle mesurée point-à-point sur une station de mesure (Alibart et al., 2012).

Ceci est à notre connaissance la première implémentation de STDP sur un disposi-

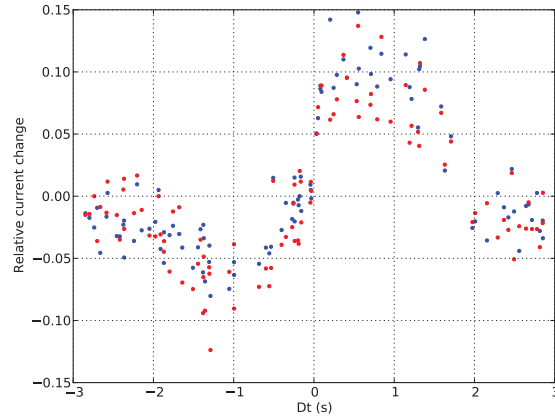


FIGURE 5.14 – Fonction d’apprentissage STDP obtenue avec la carte d’interface électronique pour deux NOMFETs (points rouge et bleu) mesurés simultanément.

tif dynamique qui satisfasse les conditions suivantes : 1) Le comportement correct est obtenu quel que soit l’état initial du dispositif, puisque l’intervalle de temps entre les impulsions pré- et post-synaptiques est aléatoire entre chaque mesure. 2) Ce comportement est très bien reproductible quelles que soient les caractéristiques des dispositifs. En effet, alors qu’il y a un facteur 10 de différence sur la conductivité entre les deux NOMFETs utilisés pour la figure 5.14, les variations relatives de conductivité sont les mêmes, ce qui signifie que la variabilité sur le comportement dynamique du NOMFET est faible. Ce comportement est dû au fait que la STDP se base sur un codage temporel, pour lequel seule la variation relative de la conductivité des NOMFETs obtenue avec les impulsions de tension et la relaxation naturelle des NPs imposent la dynamique. Ainsi, STDP nous donne un moyen de programmer de manière relativement fiable des changements de conductivité en utilisant un codage de l’information temporel avec des dispositifs peu fiables au premier abord. Un système NOMFET/STDP pourrait donc permettre d’implémenter d’autres apprentissages de type neuromorphique dans des circuits sans être trop pénalisé par les causes courantes à l’origine de la variabilité, comme les dimensions physiques, ou la reproductibilité et le contrôle des étapes de fabrication.

Ces travaux ont été publiés dans (Alibart et al., 2012).

5.4 Implémentation de STDP sur PCM

Dans cette partie, nous introduisons une nouvelle architecture faible consommation basée sur une synapse constituée de deux dispositifs PCM (la synapse 2-PCM) pour implémenter des fonctions synaptiques dans des réseaux de neurones à grande échelle. Avec cette architecture, nous simulons un réseau de neurones impulsions entièrement connectés, de type «feedforward», qui implémente notre règle STDP simplifiée. Nous évaluons l’application de détection de trajectoire de véhicules développée dans la partie 4.2.3. L’organisation de cette partie est la suivante : nous présentons d’abord un état de l’art sur les implémentations de STDP avec des dispositifs PCM, puis l’architecture que nous proposons est détaillée. Les résultats de simulation sont enfin résumés. Ceux-ci incluent une analyse détaillée des performances en apprentissage, de la robustesse, de la consommation énergétique et des perspectives d’intégration du système.

Ces travaux ont été publiés dans (Suri et al., 2011) (Bichler et al., 2012b).

5.4.1 État de l'art

Implémentation de STDP

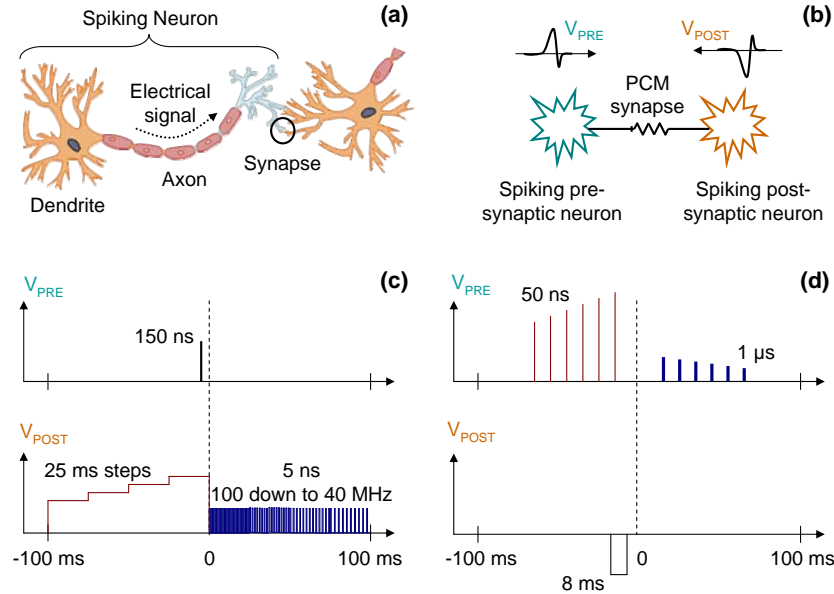


FIGURE 5.15 – (a) Neurone et synapse biologique. (b) Circuit équivalent pour une connexion synaptique. (c)-(d) Exemples d'impulsions pré- et post-synaptiques pour implémenter STDP tel que proposé dans (c) (Breitwisch et al., 2010) et (d) (Kuzum et al., 2012). Les temps indiqués correspondent à la durée des impulsions. Dans (c), l'impulsion pré-synaptique n'est pas appliquée à la seconde électrode de la PCM mais à son transistor associé (non représenté).

Plusieurs propositions d'implémentations de STDP avec des dispositifs memristifs ont été publiées ces dernières années, la plupart d'entre elles se focalisant sur des dispositifs resistifs bipolaires (Snider, 2008; Jo et al., 2010; Yu and Wong, 2010; Zamarreño-Ramos et al., 2011). La recherche porte également sur l'implémentation de l'apprentissage STDP avec de la PCM, dans le but de développer des architectures neuromorphiques pouvant être mises à l'échelle biologique. La PCM est en effet considérée comme l'une des technologies de mémoire embarquée les plus prometteuses pour la co-intégration à grande échelle avec le CMOS (Fantini et al., 2010).

Dans les implémentations actuelles de STDP avec la PCM (Breitwisch et al., 2010; Kuzum et al., 2012) (Modha and Shenoy, 2010; Jackson et al., 2011), la potentialisation synaptique, induisant la LTP, est obtenue par cristallisation tandis que la dépression synaptique, induisant la LTD, est obtenue par amorphisation du matériau chalcogénide. Des exemples d'impulsions pré- et post-synaptiques pour deux méthodes d'apprentissage différentes sont présentés figure 5.15. Dans la première implémentation (figure 5.15(c)) (Breitwisch et al., 2010), le courant n'est autorisé à circuler dans la PCM que pendant la durée de l'impulsion pré-synaptique. En fonction de son décalage temporel relatif avec l'impulsion post-synaptique, une impulsion de RESET ou une série d'impulsions de SET sont appliquées sur le dispositif, avec une amplitude dépendante de ΔT lorsque $\Delta T < 0$ ou une fréquence dépendante de ΔT lorsque $\Delta T > 0$, respectivement (avec $\Delta T = t_{\text{post}} - t_{\text{pre}}$). Sur la figure 5.15(d), l'idée de base est similaire (Kuzum et al., 2012), avec l'impulsion post-synaptique agissant comme la fonction porte au lieu de l'impulsion pré-synaptique. L'amplitude de l'impulsion est utilisée plutôt que la fréquence pour reproduire le comportement de LTP biologique.

La région amorphe dans le matériau à changement de phase peut être cristallisée par une succession d'impulsions de SET, ayant pour conséquence d'augmenter la

conductance du dispositif. Il a été montré que l'amplitude du changement relatif de conductance peut être contrôlée avec l'amplitude de l'impulsion (comme dans (Kuzum et al., 2012)) et par la largeur d'impulsion équivalente (comme dans (Breitwisch et al., 2010), où l'ampleur du changement est contrôlé par le nombre d'impulsions de SET). L'amorphisation, d'un autre côté, implique de faire fondre une région dans le matériau à changement de phase et de la refroidir abruptement («quenching»), ce qui est un processus plus gourmand en énergie et qui n'est pas progressif avec des impulsions identiques (Suri et al., 2011). La LTD induite par des impulsions de différentes amplitudes est donc absolue et non relative en valeur. Cet effet n'est donc pas apparent dans une caractérisation de STDP telle que celle montrée figure 5.15 si, pour un ΔT donné, seul le premier pas de changement de conductance est mesuré (une meilleure caractérisation de STDP pourrait être d'enregistrer un grand nombre de mesures successives avec un ΔT aléatoire, comme suggéré dans (Alibart et al., 2012)).

Règle d'apprentissage simplifiée

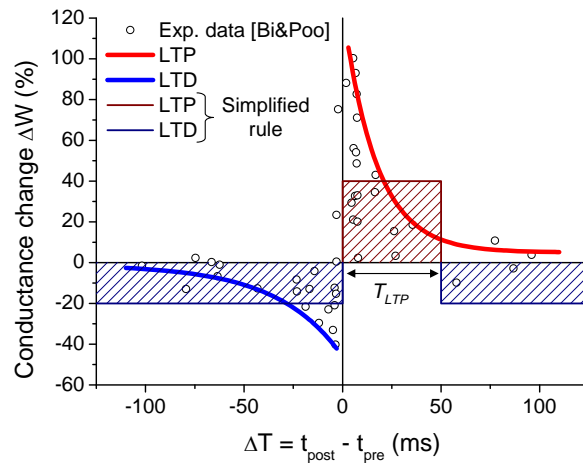


FIGURE 5.16 – Règle STDP biologique (de (Bi and Poo, 1998)) et STDP simplifiée pour l'implémentation sur PCM proposée. Dans la règle simplifiée, une synapse recevant une impulsion post-synaptique sans impulsion pré-synaptique dans la fenêtre de LTP subit une LTD, que l'impulsion pré-synaptique existe ou non.

Comme nous venons de la voir, reproduire les mesures biologiques de la règle d'apprentissage STDP, fortement dépendantes de ΔT , avec de la PCM n'est pas évident. Les impulsions de programmation doivent être adaptées à la fois en durée et en amplitude selon l'intervalle de temps précis entre les événements impulsionnels pré- et post-synaptiques. Le bénéfice de l'approche bio-mimétique (où l'on essaie de reproduire le plus fidèlement possible ce qu'il se passe en biologie) est qu'elle ne nécessite pas de faire d'hypothèse sur la manière dont seront exploitées les synapses dans le système final. En effet, si la règle de mise à jour des poids biologique de bas niveau est reproduite raisonnablement fidèlement dans un système électronique, il y a de grandes chances pour que l'apprentissage ou le traitement de haut niveau à l'œuvre dans les réseaux de neurones biologiques soit également reproductible.

Il y a toutefois des bénéfices à construire une règle STDP spécifique pour nos applications. La forme exacte de la règle d'apprentissage STDP n'est peut-être pas nécessaire pour obtenir le traitement calculatoire désiré. Par ailleurs, la courbe STDP mesurée in vitro pourrait ne pas refléter le comportement réel des neurones in vivo (Lisman and Spruston, 2010). Enfin, il n'y a pas une règle STDP unique : toute une famille de caractéristiques de mise à jour des poids en fonction de la différence temporelle entre

les impulsions pré-post a été observée (Wittenberg and Wang, 2006).

Dans ce sous-chapitre, nous proposons une nouvelle approche dans le but de résoudre des problèmes complexes, pratiques, avec la règle d'apprentissage simplifiée, qui soit efficacement implémentable avec de la PCM. Pour rappel, dans notre règle STDP, toutes les synapses d'un neurone sont affaiblies de la même ampleur lorsqu'elles reçoivent une impulsion post-synaptique, excepté pour les synapses qui ont été activées avec une impulsion pré-synaptique un court instant avant, qui sont fortement renforcées (voir la section 4.2.1). Contrairement à une synapse biologique, l'amplitude de la potentialisation ou de la dépression est indépendante de l'intervalle de temps relatif entre les impulsions pré- et post-synaptiques, comme illustré sur la figure 5.16.

5.4.2 Synapse et architecture proposées

Dans cette partie, nous décrivons le principe de fonctionnement de notre implémentation de STDP avec la PCM. Tout d'abord, la synapse 2-PCM est décrite. Ensuite, les trois modes de fonctionnement de l'architecture proposée sont détaillés : lecture, écriture et rafraîchissement.

La synapse 2-PCM

L'une des principales limitations de l'utilisation d'un seul dispositif PCM pour une synapse est l'implémentation de la LTD, qui ne peut pas être progressive avec l'amorphisation, en utilisant des impulsions identiques, comme nous l'avons déjà évoqué dans la partie 2.3.1 (Suri et al., 2011). Le courant requis pour l'amorphisation est également typiquement 5 à 10 fois plus élevé que pour la cristallisation, même pour des dispositifs de l'état de l'art (voir la figure 5.27 à la fin de ce sous-chapitre).

Pour surmonter ces difficultés, nous proposons d'implémenter à la fois la LTP et la LTD en utilisant la cristallisation, avec deux dispositifs PCM pour constituer une synapse, comme illustré sur la figure 5.17. Les deux dispositifs ont une contribution opposée à l'intégration du neurone. Quand une synapse doit être renforcée, le dispositif dit LTP subit une cristallisation partielle, ce qui augmente le poids équivalent de la synapse. De manière similaire, lorsqu'une synapse doit être affaiblie, le dispositif dit LTD est cristallisé. Comme le dispositif LTD a une contribution négative à l'intégration du neurone, le poids équivalent de la synapse est réduit. Et parce qu'une cristallisation progressive peut être obtenue avec une succession d'impulsions de tension identiques, la génération des impulsions s'en trouve grandement simplifiée.

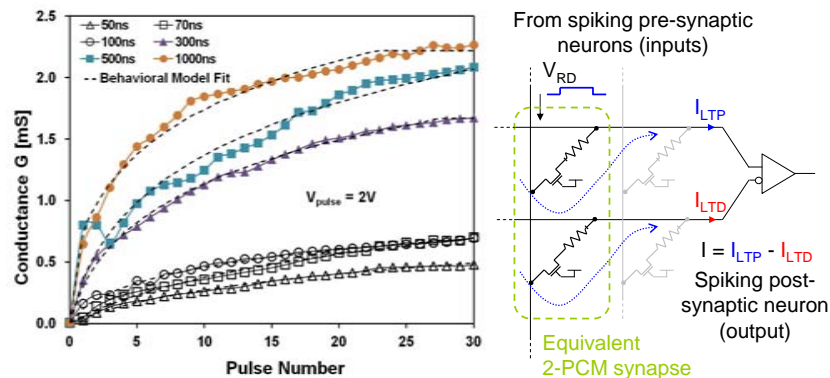


FIGURE 5.17 – Gauche : caractéristique de LTP expérimentale des dispositifs PCM Ge₂Sb₂Te₅ (GST). Pour chaque courbe, une impulsion de reset est d'abord appliquée (7 V, 100 ns), suivie par 30 impulsions de potentialisation identiques (2 V). Les lignes en pointillés correspondent au modèle comportemental utilisé dans nos simulations. Droite : principe de la synapse 2-PCM.

Avec ce système, la conductance des deux dispositifs PCM ne fait qu'augmenter lors des LTP et des LTD durant l'apprentissage, quel que soit le poids effectif de la synapse équivalente. Si aucune action n'était entreprise, les deux dispositifs finiraient par atteindre leur conductance maximale et le poids équivalent de la synapse convergerait toujours vers zéro. Ce problème est résolu en réduisant périodiquement la conductance des dispositifs tout en maintenant le poids équivalent de la synapse, avec le protocole de rafraîchissement décrit dans la partie 5.4.2. Les conséquences sur l'apprentissage de stimuli réels et le surcoût en terme de complexité et de consommation d'énergie est discuté dans la partie 5.4.3.

Opérations de lecture

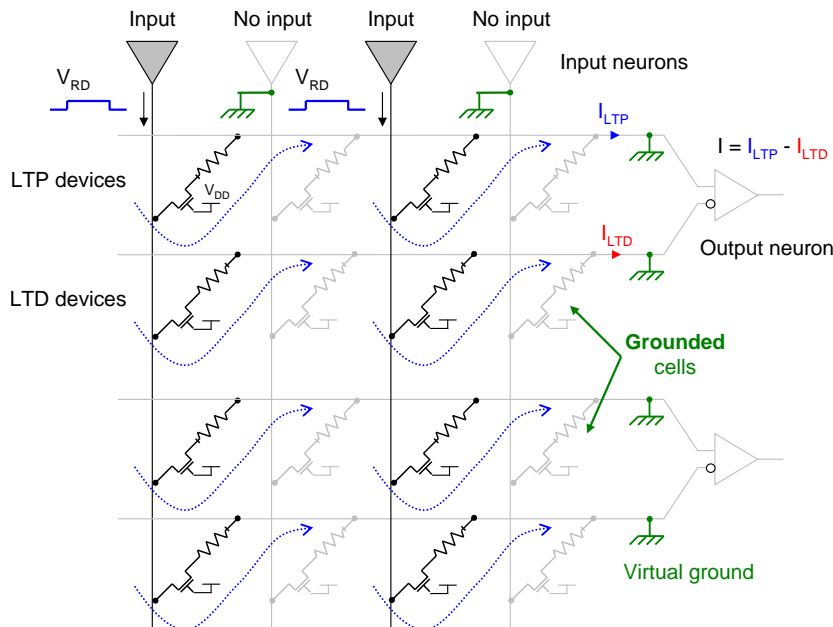


FIGURE 5.18 – Opérations de lecture. Le courant passant dans les dispositifs PCM LTP et LTD est intégré dans le neurone de sortie, avec une contribution positive et négative respectivement.

L'opération de lecture décrite ici est le processus de fonctionnement normal du réseau entre deux activations des neurones de sortie. Lorsqu'un neurone d'entrée reçoit un événement, celui-ci génère une impulsion de tension de faible amplitude propagée vers l'ensemble des neurones de sortie via ses synapses. Le courant résultant entrant dans chaque neurone de sortie est la différence entre les courants provenant des dispositifs LTP et ceux des dispositifs LTD, comme montré sur la figure 5.18. L'amplitude et la durée de l'impulsion de lecture peuvent être minimum, tant qu'elle permet une lecture raisonnablement précise de la résistance faible champ de la PCM. Les neurones de sortie sont de type «Leaky integrate-and-fire» (LIF) (Livi and Indiveri, 2009; Joubert et al., 2011). Lorsque le courant intégré atteint le seuil du neurone, le réseau entre dans un mode d'écriture pour la mise à jour des poids synaptiques avec STDP, comme décrit dans la suite.

A chaque fois qu'un neurone d'entrée est activé, celui-ci entre ou ré-entre dans un état interne de LTP pour la durée de la fenêtre de LTP de STDP, T_{LTP} (la valeur effectivement utilisée dans les simulations est donnée dans le tableau 5.1).

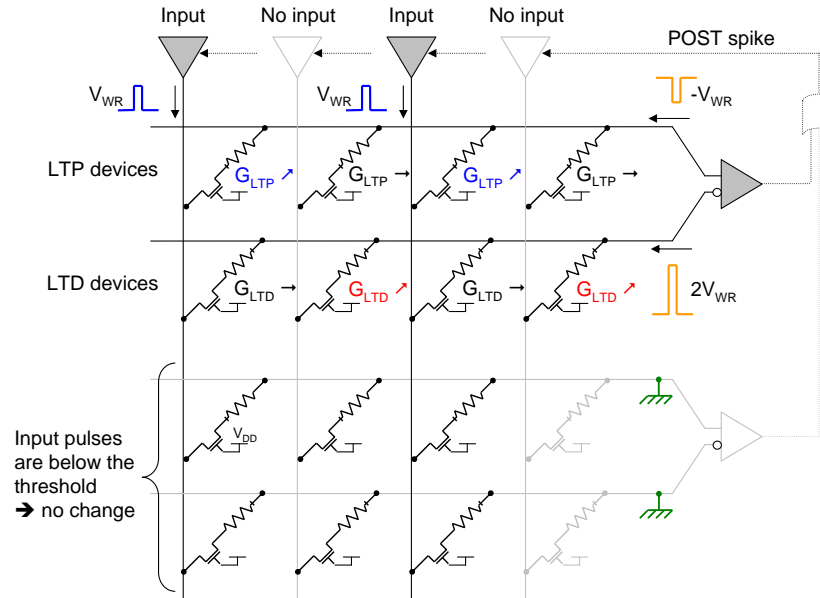


FIGURE 5.19 – Opérations d’écriture pour la règle STDP simplifiée. Pour un dispositif PCM donné, $G \nearrow$ représente une augmentation de la conductance (et donc une cristallisation partielle du dispositif), alors que $G \rightarrow$ signifie pas de changement de la conductance.

Opérations d’écriture

Quand un neurone de sortie se déclenche, il transmet un signal de post impulsion à tous les neurones d’entrée, leur signifiant les opérations d’écriture. Dans les opérations d’écriture, les neurones d’entrée génèrent une impulsion de LTP d’amplitude V_{WR} de manière à ce que $V_{WR} < V_{SET} < 2V_{WR}$, uniquement lorsqu’ils sont dans l’état LTP. Le neurone de sortie génère une impulsion de rétroaction négative $-V_{WR}$ pour les dispositifs LTP et une impulsion de rétroaction positive $2V_{WR}$ pour les dispositifs LTD. Lorsqu’une impulsion de LTP interagit avec une impulsion de rétroaction, la tension effective aux bornes du dispositif LTP est $2V_{WR} > V_{SET}$ et la tension aux bornes du dispositif LTD est $V_{WR} < V_{SET}$. La conductance du dispositif LTP est donc augmentée. S’il n’y a pas d’impulsion de LTP pour une entrée donnée, cela signifie que la différence temporelle pré-post n’est pas dans la fenêtre de LTP et donc que la conductance du dispositif LTD doit être augmentée d’après notre règle STDP simplifiée. Cela est effectivement le cas, puisque la tension aux bornes du dispositif LTP est $-V_{WR} > -V_{SET}$ et la tension aux bornes du dispositif LTD est $2V_{WR} > V_{SET}$ (voir figure 5.19).

Des transistors de sélection ne sont pas nécessaires pour les opérations d’écriture, car l’amplitude de l’impulsion d’entrée de LTP est sous le seuil de SET de la PCM, les poids synaptiques des autres neurones de sortie ne sont donc pas affectés. Les impulsions de LTP peuvent néanmoins significativement altérer la valeur de l’intégration dans d’autres neurones de sortie. Ceci n’est pas un problème dans l’architecture proposée puisque de l’inhibition latérale est implémentée : lorsqu’un neurone se déclenche, l’intégration des autres neurones est désactivée pour une période d’inhibition $T_{inhibit}$.

Opérations de rafraîchissement

Du fait que la conductance des dispositifs PCM augmente progressivement durant l’apprentissage, un mécanisme de rafraîchissement est introduit pour réduire la conductance des dispositifs LTP et LTD tout en maintenant le poids de la synapse équivalente inchangé. Le principe de l’opération de rafraîchissement est montré sur la figure 5.20. Régulièrement, les dispositifs sont tous les deux programmés dans l’état RESET et ce-

lui qui avait la conductance la plus élevée subit une série d'impulsions de SET jusqu'à ce que le poids équivalent soit à nouveau atteint. Comme l'un des dispositifs reste à sa conductance minimale, ce mécanisme permet une évolution continue des poids.

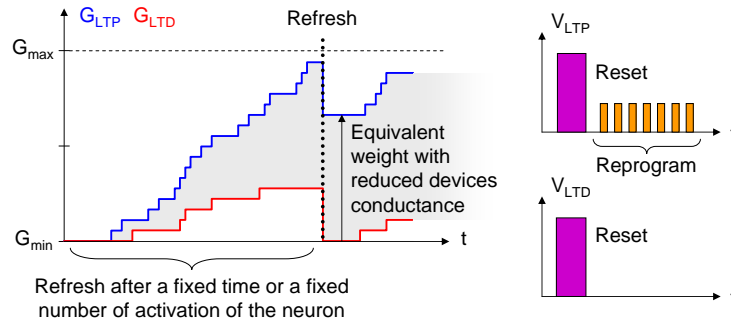


FIGURE 5.20 – Principe du rafraichissement : les deux dispositifs constituant la synapse sont programmés dans l'état RESET et celui qui avait la conductance la plus élevée est reprogrammé de manière à ce que le poids équivalent de la synapse reste inchangé.

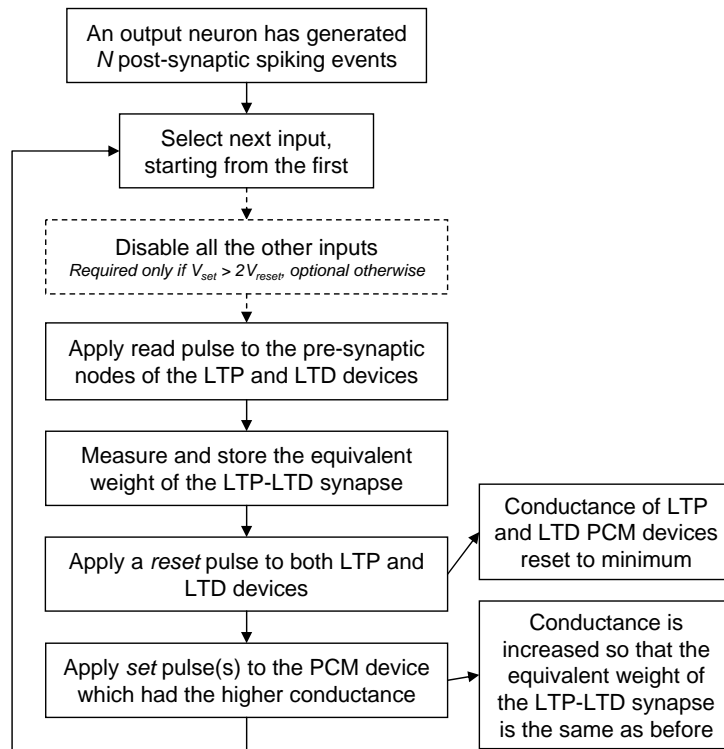


FIGURE 5.21 – Description des opérations de rafraichissement.

Connaissant le nombre moyen de pas de conductance N atteignable pour une technologie PCM, une amplitude et une durée d'impulsion de SET données, une opération de rafraichissement n'est nécessaire qu'après N potentialisations ou N dépressions de la synapse (le cas qui arrive en premier, en supposant que l'un des deux dispositifs est initialement à sa conductance minimale). Ainsi, les neurones de sortie peuvent initier une opération de rafraichissement sur leurs synapses après un nombre fixe d'activations, qui serait N dans le pire des cas. Bien qu'un mécanisme aussi simple implique certainement un nombre substantiel de RESET inutiles, puisque peu de synapses subissent N potentialisations ou N dépressions d'affilé, cela a l'avantage de ne pas nécessiter l'observation continue de l'état des dispositifs LTP et LTD. N peut être élevé (une valeur approchant 100 a été montrée dans (Breitwisch et al., 2010)), réduisant ainsi le

surcrot en temps/énergie au minimum. Les simulations ont montré que même $N = 10$ n'implique qu'un surcrot marginal pour le système sur une expérience d'apprentissage d'un stimulus réel avec pratiquement 2 000 000 de synapses (voir partie suivante). Cela n'inclut toutefois pas le surcrot lié au CMOS.

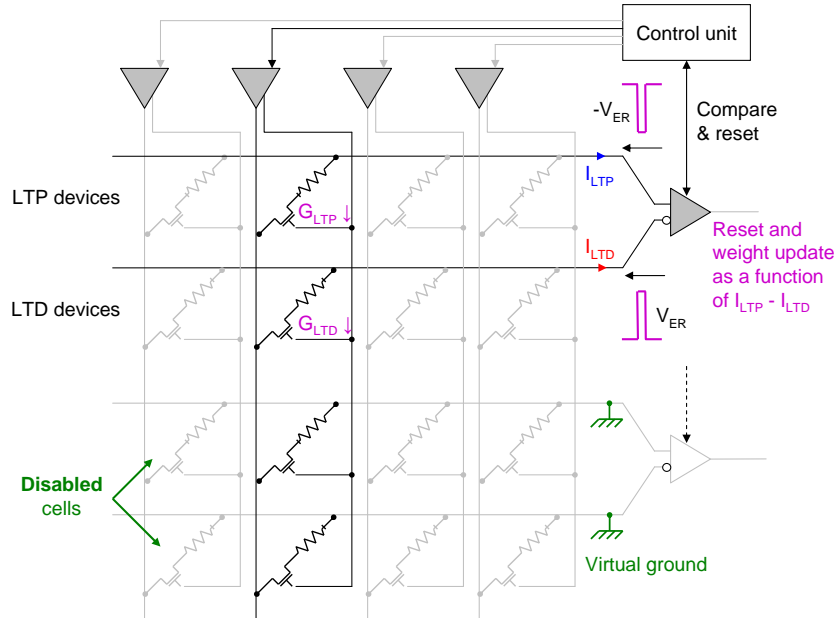


FIGURE 5.22 – Opérations de rafraîchissement : génération des impulsions de RESET pour réinitialiser la conductance des dispositifs LTP et LTD au minimum lorsque $V_{RESET} > 2V_{SET}$.

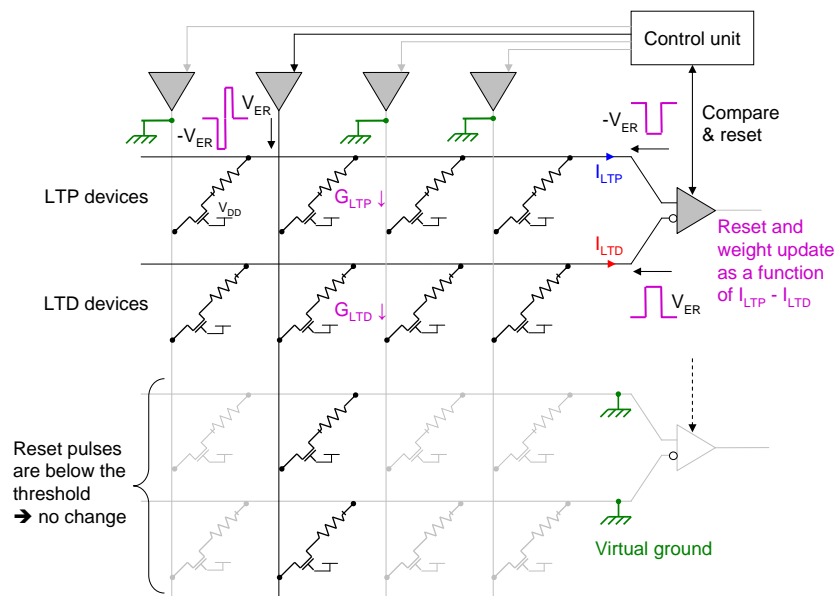


FIGURE 5.23 – Opérations de rafraîchissement sans transistor de sélection : génération des impulsions de RESET pour réinitialiser la conductance des dispositifs LTP et LTD au minimum lorsque $V_{RESET} < 2V_{SET}$. Le dispositif LTD (respectivement LTP) est RESET lorsque la partie de l'impulsion d'effacement négative $-V_{ER}$ (respectivement positive V_{ER}) provenant du neurone d'entrée recouvre l'impulsion d'effacement post-synaptique.

Les opérations de rafraîchissement sont décrites sur le diagramme de la figure 5.21. Les synapses sont lues, effacées et reprogrammées séquentiellement. Les autres neurones sont désactivés durant le processus. Afin de fortement amorphiser la PCM, une

impulsion de RESET d'amplitude V_{RESET} doit être appliquée au dispositif, comme illustré sur la figure 5.22. Si $V_{\text{RESET}} < 2V_{\text{SET}}$, une tension de V_{RESET} aux bornes de la PCM peut être obtenue par l'interaction de deux impulsions d'amplitude V_{ER} telle que $V_{\text{ER}} < V_{\text{SET}} < 2V_{\text{ER}}$, comme illustré sur la figure 5.23. Dans ce cas, la tension aux bornes des autres synapses du «crossbar» est toujours inférieure au seuil de SET et leur conductance n'est pas affectée.

Ainsi, si la condition $V_{\text{RESET}} < 2V_{\text{SET}}$ est vraie, aucun transistor de sélection n'est nécessaire pour les opérations de rafraîchissement. Cette condition est généralement vérifiée pour les PCM de l'état de l'art (Pirovano et al., 2007). Comme ni les opérations de lecture, ni les opérations d'écriture n'en nécessitent un, les dispositifs de sélection peuvent être complètement éliminés. Cela pourrait théoriquement autoriser la plus grande densité d'intégration possible avec la PCM dans un «crossbar» et libérer la couche CMOS sous-jacente pour l'intégration des neurones.

Si $V_{\text{RESET}} > 2V_{\text{SET}}$, la tension V_{RESET} ne peut pas être obtenue avec deux impulsions d'amplitude inférieure à V_{SET} . Des dispositifs de sélection sont donc nécessaires pour désactiver les autres dispositifs PCM et empêcher leur conductance d'être modifiée, lorsque la synapse 2-PCM en cours de rafraîchissement est RESET, comme montré sur la figure 5.22 (les PCMs désactivés sont grisées).

5.4.3 Résultats de simulation

Dans cette partie, nous présentons les résultats d'une simulation d'un apprentissage à grande échelle de notre implémentation de STDP avec des synapses 2-PCMs. Nous utilisons le simulateur événementiel C++ Xnet. Le modèle comportemental asymétrique (Querlioz et al., 2011b) est utilisé pour interpoler la caractéristique de LTP des PCM GST (figure 5.17) et GeTe. Les opérations de rafraîchissement telles que décrites précédemment sont également simulées.

Topologie du réseau et stimuli d'entrée

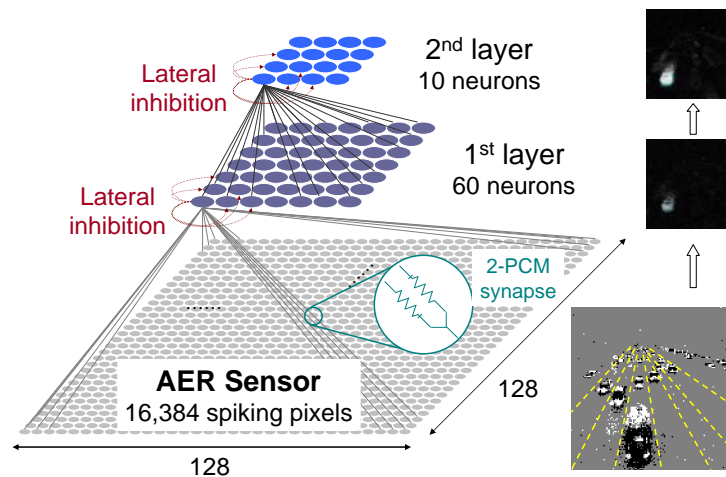


FIGURE 5.24 – Topologie du réseau utilisé pour la simulation. Il est complètement connecté et chaque pixel du capteur AER de 128×128 pixels est connecté à chaque neurone de la première couche avec deux synapses, recevant respectivement les événements de changement positif et négatif d'intensité lumineuse.

La figure 5.24 montre la topologie du réseau de neurones impulsionnels «feedforward» à deux couches simulé. C'est un réseau complètement connecté, avec 60 neurones sur la première couche et 10 neurones sur la seconde. La couche du dessous représente

le capteur AER de 128×128 pixels (Lichtsteiner et al., 2008). Un pixel génère un évènement à chaque fois que le changement relatif d'intensité lumineuse qu'il reçoit atteint un seuil positif ou négatif. Ainsi, selon le signe du changement d'intensité, les évènements peuvent être soit de type ON, soit de type OFF, correspondant respectivement à une augmentation ou une diminution d'intensité lumineuse. Il y a deux synapses par pixel, une par type d'évènement. Le nombre total de synapses dans le système est de $2 \times 128 \times 128 \times 60 + 60 \times 10 = 1\,966\,680$ et donc le nombre de dispositifs PCM est de 3 933 360 (2 PCM/synapse). Les neurones sont standards LIF avec une période réfractaire T_{refrac} . Lorsqu'un neurone se déclenche, il désactive les autres neurones pendant une période $T_{inhibit}$, durant laquelle aucune impulsion d'entrée n'est intégrée.

Le stimulus utilisé dans les simulations suivantes a été enregistré avec le capteur TMPDIFF128. Il représente des véhicules passant sous un pont sur l'autoroute 210 à Pasadena. La séquence dure 78.5 s et contient un total de 5.2M évènements, avec une fréquence moyenne de 66.1k évènements par seconde. L'encart en bas à droite de la figure 5.24 montre un rendu de la séquence, où les voies de circulation ont été mises en évidence. Avec STDP, le réseau de neurones est capable d'extraire n'importe quel motif répétitif dans le stimulus de manière non supervisée. Dans le cas présent, les motifs répétitifs appris correspondent aux trajectoires des véhicules, qui peuvent être utilisés pour compter les véhicules sur chaque voie. Pour rappel, nous avons détaillé ce mécanisme dans la partie 4.2.3.

Les paramètres des neurones pour les deux couches sont donnés dans la tableau 5.1, pour des réseaux simulés avec de la PCM GST et de la PCM GeTe. I_{thres} est le seuil du neurone, exprimé en siemens (S) pour le rendre indépendant de la tension et de la durée de l'impulsion de lecture. τ_{leak} est la constante de temps de fuite du neurone. Les paramètres des neurones sont adaptés à la dynamique des stimuli, correspondant à l'activité évènementielle moyenne générée par les véhicules en bas de la rétine (où l'activité est maximale à cause de la vue en perspective). Nous avons donné plus d'information sur la signification et l'optimisation de ces paramètres pour l'apprentissage au chapitre traitant des applications, notamment dans la partie 4.2.2. N est le nombre d'activations requis d'un neurone de sortie pour initier les opérations de rafraichissement. LTP/LTD est la force relative (ou l'évolution du poids équivalent) induite par la LTP comparée à la LTD. Le rapport LTP/LTD de 2 utilisé dans nos simulations assure que les synapses renforcées de façon répétitive convergent vers leur poids maximal suffisamment rapidement pour que le neurone devienne sélectif à une voie de circulation. Cela peut être implémenté en ajoutant un gain en courant de 2 sur l'entrée LTP des neurones.

TABLE 5.1 – Paramètres des neurones pour l'apprentissage. Un jeu de paramètre différent est utilisé selon le matériau PCM.

Paramètre	GST		GeTe	
	1ère couche	2ème couche	1ère couche	2ème couche
I_{thres}	2.49 S	0.00437 S	2.50 S	0.00431 S
T_{LTP}	7.59 ms	7.12 ms	11.5 ms	12.9 ms
T_{refrac}	554 ms	410 ms	524 ms	393 ms
$T_{inhibit}$	15.7 ms	56.5 ms	11.8 ms	70.9 ms
τ_{leak}	100 ms	821 ms	115 ms	714 ms
N	30		10	
LTP/LTD	2.0		2.0	

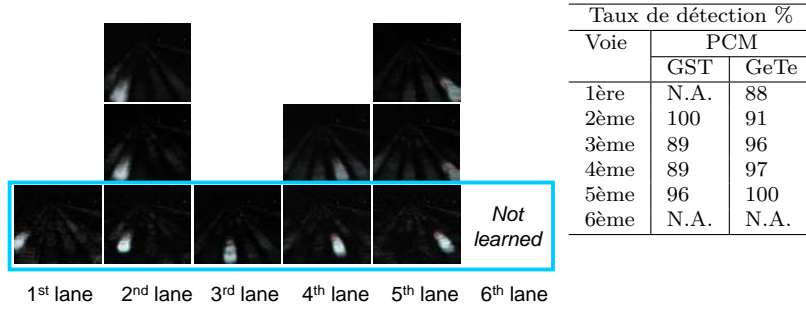


FIGURE 5.25 – Gauche : détection des véhicules sur les 6 voies de circulation par les 10 neurones de sortie de la seconde couche. Chaque bloc montre la carte de sensibilité pour un neurone de sortie (ici pour la PCM GeTe). De haut en bas le taux de détection augmente. Les voies 2 et 5 sont apprises par 3 neurones chacune, tandis que la voie 6 n’est apprise par aucun neurone. Droite : taux de détection pour les 5 meilleurs neurones, pour chaque voie.

Performances d’apprentissage et robustesse

La figure 5.25 montre le résultat de l’apprentissage pour le jeu de données AER. Les neurones sont sélectifs à une unique voie de circulation après seulement 8 présentations de la séquence, correspondant approximativement à une durée temps réel de trafic de 10 minutes. L’apprentissage STDP et l’inhibition latérale peuvent être complètement désactivés après l’apprentissage pour une détection continue des véhicules. Les neurones de sortie de la seconde couche sont capables de détecter respectivement les véhicules sur 4 voies de circulation sur 6 pour le système basé sur des synapses PCM GST et 5 voies sur 6 pour le système basé sur des synapses PCM GeTe. La sixième voie de circulation n’est jamais apprise, car elle se trouve tout à droite de la rétine et les véhicules activent moins de pixels sur cette trajectoire que sur les autres voies. Sur les voies apprises, le taux de détection moyen est supérieur à 92%, avec aucun faux positif (c’est-à-dire que les neurones ne se déclenchent qu’une seule fois par véhicule et qu’ils ne se déclenchent jamais pour un véhicule passant sur une voie différente de celle qu’ils ont apprise). Des statistiques sur l’apprentissage sont données dans le tableau 5.2 : la fréquence de mise à jour des poids synaptiques (ou fréquence post-synaptique) est de l’ordre de 0.1 Hz et la fréquence pré-synaptique moyenne est autour de 2 Hz. Les fréquences moyennes sont similaires pour les deux couches.

TABLE 5.2 – Statistiques d’apprentissage, sur l’ensemble de la durée d’apprentissage (8×85 = 680 s). Le nombre d’impulsions de SET inclut à la fois les impulsions d’écriture pour l’apprentissage et les impulsions supplémentaires nécessaires pour reprogrammer le poids synaptique équivalent durant les opérations de rafraîchissement (figure 5.22).

	/dispo.	/dispo. (max.)	/dispo./s	Total
GST (impulsions LTP 2 V / 300 ns)				
Impulsions de lecture	1265	160 488	1.9	4 975 830 080
Impulsions de SET	106	430	0.16	416 334 080
Impulsions de RESET	4.2	7	0.0062	16 585 048
GeTe (impulsions LTP 1.5 V / 100 ns)				
Impulsions de lecture	1265	160 488	1.86	4 975 848 000
Impulsions de SET	190	740	0.28	748 120 539
Impulsions de RESET	20	37	0.030	79 971 200

La fréquence moyenne des impulsions de potentialisation par dispositif est environ 25 fois plus élevée que la fréquence des impulsions de RESET pour le système basé sur des PCM GST, et environ 10 fois plus élevée pour le système basé sur des PCM GeTe.

Cela est cohérent avec le fait que les opérations de rafraichissement sont initiées après 30 activations pour un neurone de sortie donné pour GST et seulement 10 activations pour GeTe. Comme cela a déjà été mentionné, ce résultat suggère que l'efficacité du système peut encore être améliorée en choisissant le bon matériau à changement de phase avec la fenêtre de conductance optimale pour maximiser le nombre de niveaux de conductance atteignables avec une série d'impulsions de SET identiques. L'approche proposée de stocker le maximum d'informations synaptiques avec la cristallisation de la PCM, avec une fréquence d'écriture d'environ 2 Hz par dispositif PCM, réduit également l'impact du « drift » de la résistance PCM (Fantini et al., 2010), considéré comme une limitation sérieuse de la programmation PCM multi-niveaux.

Afin d'évaluer la tolérance à la variabilité de notre système neuromorphique, le réseau de neurone simulé inclut une dispersion pessimiste de 20% (c'est-à-dire un écart-type sur chaque paramètre valant 20% de leur valeur moyenne) pour tous les paramètres du modèle asymétrique utilisé pour la PCM (Querlioz et al., 2011b) (G_{min} , G_{max} , α and β). Un exemple de caractéristiques de PCM pour une série d'impulsions de SET identiques est montré sur la figure 5.26, pour 100 différents jeux de paramètres obtenus en ajoutant une dispersion de 20% sur les données extraites à partir de l'interpolation. Dans nos simulations, les paramètres d'un dispositif PCM sont modifiés chaque fois que le dispositif subit un RESET. La dispersion de 20% inclut donc à la fois la variabilité entre dispositifs et la variabilité entre des séquences consécutives de SET sur le même dispositif.

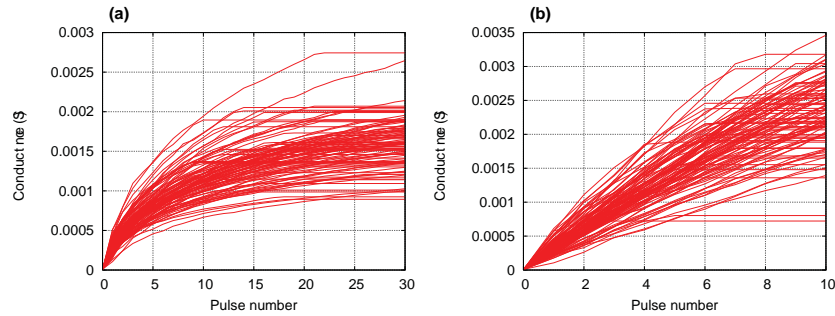


FIGURE 5.26 – Variabilité simulée pour de la PCM GST (a) et GeTe (b). Les courbes montrent l'augmentation de la conductance en fonction du nombre d'impulsions de SET, pour 100 différents jeux de paramètres, obtenus en appliquant une dispersion de 20% (écart-type par rapport à la valeur moyenne) sur les valeurs extraites à partir de l'interpolation.

Consommation d'énergie et mise à l'échelle

A partir des statistiques d'apprentissage du tableau 5.2, nous avons fait une estimation grossière de l'énergie consommée pour la programmation des dispositifs PCM GST :

$$E_{total} = E_{SET} \cdot N_{SET \text{ pulses}} + E_{RESET} \cdot N_{RESET \text{ pulses}} \quad (5.2)$$

$$\text{avec } E_{SET} \approx V_{SET} \cdot I_{SET} \cdot t_{SET} \quad (5.3)$$

$$\text{avec } E_{RESET} \approx V_{RESET} \cdot I_{RESET} \cdot t_{RESET} \quad (5.4)$$

Avec les tensions et les courants de SET et de RESET mesurés sur les dispositifs GST et $t_{SET} = 30 \text{ ns}$, $t_{RESET} = 50 \text{ ns}$, $E_{SET} \approx 121 \text{ pJ}$ et $E_{RESET} \approx 1552 \text{ pJ}$. En utilisant ces valeurs, la consommation d'énergie synaptique pour l'apprentissage est de $112 \mu\text{W}$. Nous n'avons pas inclus l'énergie pour la lecture dans le calcul, celle-ci étant

négligeable. En effet, dans le pire des cas, l'énergie totale de lecture serait la suivante :

$$E_{\text{total read}} = E_{\text{read}_{\text{max}}} \cdot N_{\text{read pulses}} \quad (5.5)$$

$$\text{avec } E_{\text{read}_{\text{max}}} \approx V_{\text{read}}^2 \cdot G_{\text{max}} \cdot t_{\text{read}} \quad (5.6)$$

Avec $V_{\text{read}} = 0.1$ V et $t_{\text{read}} = 10$ ns, nous estimons $E_{\text{read}_{\text{max}}} \approx 0.17$ pJ et $E_{\text{total read}} \approx 0.8$ μ W. Ce calcul n'inclut pas le circuit CMOS pour les neurones et néglige également l'énergie supplémentaire requise pour charger les capacités des lignes dans le «crossbar», qui peut être conséquente dans les technologies avancées.

La figure 5.27 montre qu'en moyenne, le courant requis pour le RESET et le SET évolue pratiquement linéairement avec la surface du matériau à changement de phase. Le tableau 5.3 donne des estimations de la consommation d'énergie synaptique pour plusieurs dispositifs dont les caractéristiques ont été publiées. Avec des technologies PCM extrêmement avancées, une consommation énergétique aussi faible que 100 nW semble être atteignable pour les ~ 2 millions de synapses avec un apprentissage STDP en continu. Si l'apprentissage n'intervenait que durant des périodes limitées dans le temps, la consommation d'énergie pourrait être de plusieurs ordres de grandeur plus faible.

TABLE 5.3 – Statistiques énergétiques et consommation synaptique pour le cas décrit dans le tableau 5.2, en utilisant des valeurs de tension et de courant extraites de la littérature.

Technologie PCM	E_{RESET} (pJ)	E_{SET} (pJ)	Puissance (μ W)
Ces travaux (PCM GST)	1552	121	112
(Liang et al., 2011)	1.2	0.045	0.056
(Xiong et al., 2011)	0.1	0.03	0.02
(Pirovano et al., 2007)	24	4.9	3.6
(Im et al., 2008)	5.6	0.9	0.68

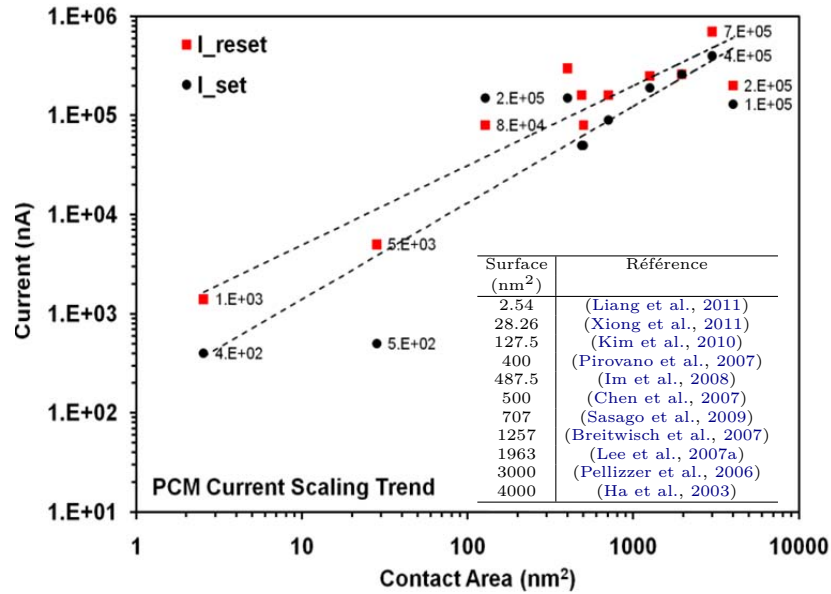


FIGURE 5.27 – Évolution des courants de RESET et de SET pour différentes technologies PCM (valeurs extraites de la littérature). L'interpolation est également montrée (exposants : ~ 1.2 pour SET et ~ 0.9 pour RESET).

Avec une fréquence moyenne de SET/RESET par dispositif de l'ordre de 1 Hz, un apprentissage continu sur 3 ans nécessiterait une endurance de 10^8 cycles, ce qui est fa-

cilement accessible avec de la PCM (Kim et al., 2010). La dégradation des performances serait progressive à mesure que les dispositifs se détériorent, grâce au haut niveau de tolérance de ce type de réseau de neurones.

5.5 Apprentissage STDP stochastique sur CBRAM

Nous avons vu dans le chapitre «Technologies mémoires émergentes et memristivité» que la CBRAM posait un certain nombre de problèmes pour une utilisation en tant que synapse (section 2.4.1). La programmation multi-niveaux nécessite l'utilisation d'un transistor de sélection pour contrôler le courant. Cette solution a été proposée pour implémenter STDP avec des trains d'impulsions pré- et post-synaptiques d'amplitude décroissante (Yu and Wong, 2010). Le principe est illustré sur la figure 5.28. Dans le cas d'une LTP (cas $\Delta t > 0$ sur la figure), la première impulsion négative post-synaptique s'additionne aux bornes du dispositif avec une impulsion pré-synaptique d'amplitude plus ou moins grande selon Δt , émulant ainsi la modulation de l'amplitude de la LTP biologique en fonction du temps. Les autres impulsions constituant les trains pré- et post-synaptiques se compensent partiellement et restent sous le seuil du dispositif CBRAM. Le fonctionnement est similaire dans le cas de la LTD.

Cette proposition d'implémentation présente toutefois de sérieuses limitations. La plus importante est que la variation de conductance de la CBRAM n'est pas cumulative. Le mécanisme de LTP/LTD proposé programme une nouvelle conductance qui ne dépend pas de la précédente valeur de celle-ci. Une limitation similaire avait déjà été évoquée dans le cas de la LTD avec la PCM (section 2.3.1). Contourner cette limitation nécessiterait de lire l'état du dispositif avant chaque changement de conductance, et d'adapter l'amplitude des trains d'impulsions en conséquence. L'utilisation d'un train d'impulsions accroît également la consommation du système, dont la part due à la charge des capacités des lignes dans le «crossbar» peut être significative. Enfin, nos caractérisations sur la CBRAM ont montré que la programmation multi-niveaux pour la LTD, qui nécessite de contrôler la dissolution du filament, est difficile à obtenir.

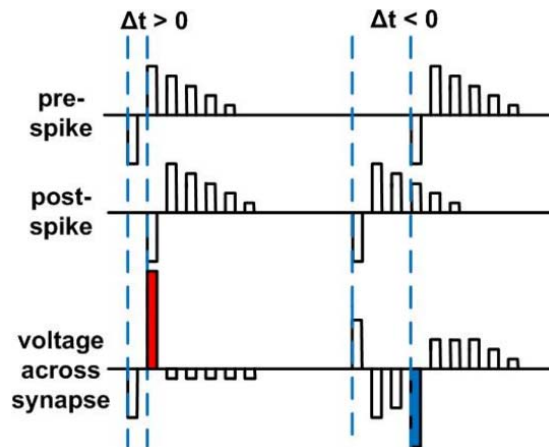


FIGURE 5.28 – Proposition d'implémentation de STDP avec de la CBRAM (Yu and Wong, 2010). Il est possible de programmer différents niveaux de conductance en contrôlant le courant dans la CBRAM avec un transistor de sélection.

Pour éliminer ces difficultés, nous proposons d'utiliser la CBRAM en tant que synapse binaire, avec une règle d'apprentissage STDP stochastique. L'approche que nous proposons peut s'appliquer à n'importe quel autre dispositif binaire bipolaire. Dans le cas de la CBRAM et potentiellement d'autres dispositifs de type RRAM (Jo et al., 2009b), ce type d'apprentissage stochastique peut tirer profit de la stochasticité in-

trinsèque des dispositifs pour diminuer la consommation du système en utilisant des conditions de programmation faibles, voire pour éliminer la nécessité de faire appel à des générateurs pseudo-aléatoires, comme nous le verrons dans la suite.

Ces travaux ont été publiés dans (Suri et al., 2012) (Querlioz et al., 2012b).

5.5.1 Synapse et architecture proposées

Pour de l'apprentissage stochastique, une synapse est constituée d'un ou de plusieurs dispositifs CBRAM qui sont utilisés individuellement en tant que dispositif mémoire binaire. Les dispositifs CBRAM ont été fabriqués par Altis Semiconductor et caractérisés au Laboratoire d'électronique des technologies de l'information du Commissariat à l'énergie atomique et aux énergies alternatives (CEA) (CEA LETI). Pour nos simulations, nous avons établi un modèle binaire dans lequel la distribution des états ON et OFF est modélisée par une loi log-normale. Le modèle est également probabiliste, car il tient compte du fait que le taux de succès du SET et du RESET n'est pas toujours de 100% selon les conditions de programmation utilisées. L'ajustement du modèle se fait pour un seul jeu de conditions de programmation (amplitude et durée des impulsions) pour le SET et le RESET. Les paramètres du modèle utilisés pour les simulations de la partie suivante sont donnés sur la figure 5.29.

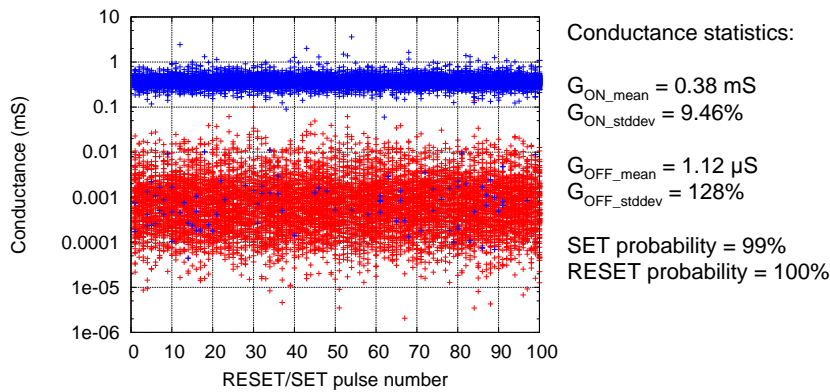


FIGURE 5.29 – Caractéristiques de la modélisation binaire de la CBRAM. La figure représente les variations sur les niveaux haut et bas de la conductance pour 100 cycles de SET/RESET sur 64 dispositifs. Le modèle utilise une distribution log-normale, qui s'ajuste le mieux aux données expérimentales. Avec les conditions de programmation utilisées, la probabilité de passage à l'état ON avec un SET n'est que de 99%.

La règle STDP simplifiée que nous avons utilisée jusqu'à présent est exprimée sous une forme probabiliste sur la figure 5.30. Le principe reste le même : par rapport à la règle biologique, toutes les synapses n'ayant pas été activées dans la fenêtre de LTP subissent une LTD. Dans sa forme probabiliste, la variation du poids via une LTP ou une LTD n'est pas relative à l'état précédent de la synapse et est stochastique. L'utilisation de plusieurs dispositifs binaires pour une synapse est fonctionnellement équivalent à un seul dispositif multi-niveaux, à condition que les probabilités de passage à l'état ON/OFF de chaque dispositif soient indépendantes les unes des autres.

Le principe de programmation des synapses est similaire à celui déjà présenté dans la première partie de ce chapitre. La figure 5.31 résume les quatre cas possibles pour la programmation des synapses, lors d'une activation d'un neurone de sortie, sur quatre neurones d'entrée d'une matrice. De gauche à droite :

- L'entrée a été activée dans la fenêtre de LTP, la synapse doit donc être renforcée. Cette potentialisation est probabiliste, comme illustré sur la figure 5.30. Le neurone d'entrée a donc une probabilité P_{eLTP} de générer une impulsion de programmation. Pour ce faire, un générateur pseudo-aléatoire binaire avec une probabilité

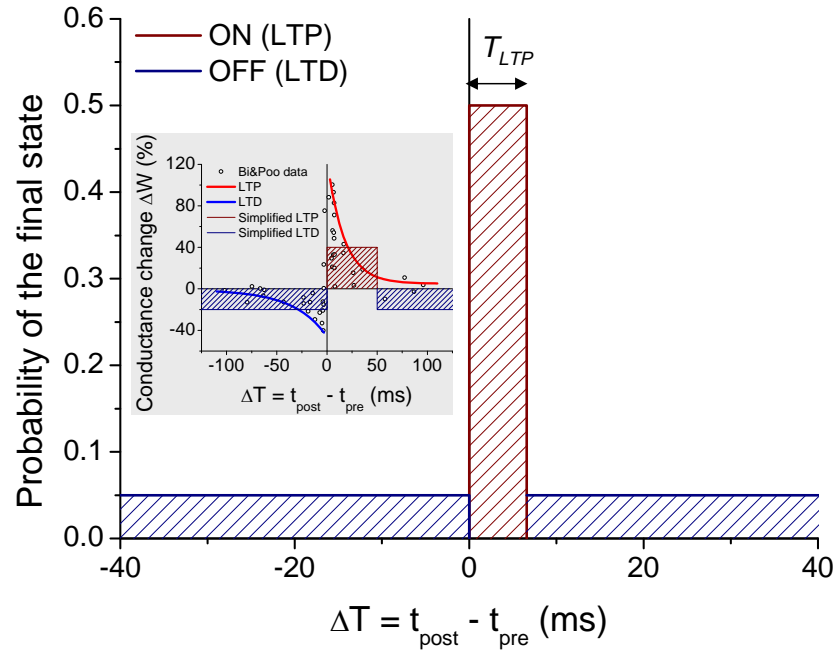


FIGURE 5.30 – Règle d'apprentissage STDP stochastique. Une LTP/LTD ne correspond plus à une variation relative du poids de la synapse, mais à une probabilité de changer son poids pour le poids maximal (pour la LTP, en rouge) ou pour un poids minimal (pour la LTD, en bleu).

Pe_{LTP} de sortir un «1» logique est utilisé. Dans le premier cas de la figure, le bit de sortie du générateur est à «1» et une impulsion de programmation est générée. Celle-ci interagit avec la première partie de l'impulsion post-synaptique pour entraîner la commutation de la synapse à l'état ON ;

- L'entrée suivante n'a pas été activée dans la fenêtre de LTP, la synapse doit donc être affaiblie. Le neurone a une probabilité Pe_{LTD} de générer une impulsion de programmation. La probabilité Pe_{LTD} n'étant pas nécessairement identique à Pe_{LTP} , un deuxième générateur pseudo-aléatoire est nécessaire. Dans le deuxième cas présenté sur la figure, la sortie du générateur est à «0» et le poids de la synapse n'est pas modifié ;
- La troisième entrée a été activée dans la fenêtre de LTP, mais la synapse n'est pas programmée ;
- La quatrième entrée n'a pas été activée dans la fenêtre de LTP et une impulsion de programmation est générée. Celle-ci interagit avec la deuxième partie de l'impulsion post-synaptique pour entraîner la commutation de la synapse à l'état OFF.

La figure 5.32 présente une implémentation possible d'un générateur de nombres pseudo-aléatoires – «Pseudorandom Number Generator» (PRNG) biaisé, avec une probabilité de générer un «1» logique réglable. Celui-ci est constitué de plusieurs registres à décalage à rétroaction linéaire – «Linear Feedback Shift Registers» (LFSRs) indépendants (Brglez et al., 1989). Sachant que la probabilité d'obtenir un «1» logique en sortie d'un LFSR est de 0.5, la probabilité d'obtenir un «1» logique en sortie d'une porte ET sur deux LFSRs indépendants est de 0.25. Toujours avec l'hypothèse d'indépendance, les probabilités sont multipliées en sortie d'une porte ET et additionnées en sortie d'une porte OU. Il est ainsi possible produire une distribution de «1» logique de précision arbitraire.

Le circuit présenté à la figure 5.31 permet d'implémenter une STDP stochastique pour n'importe quel dispositif memristif binaire bipolaire, de manière extrinsèque, c'est-à-dire sans que la commutation du dispositif en lui-même ne nécessite d'être probabi-

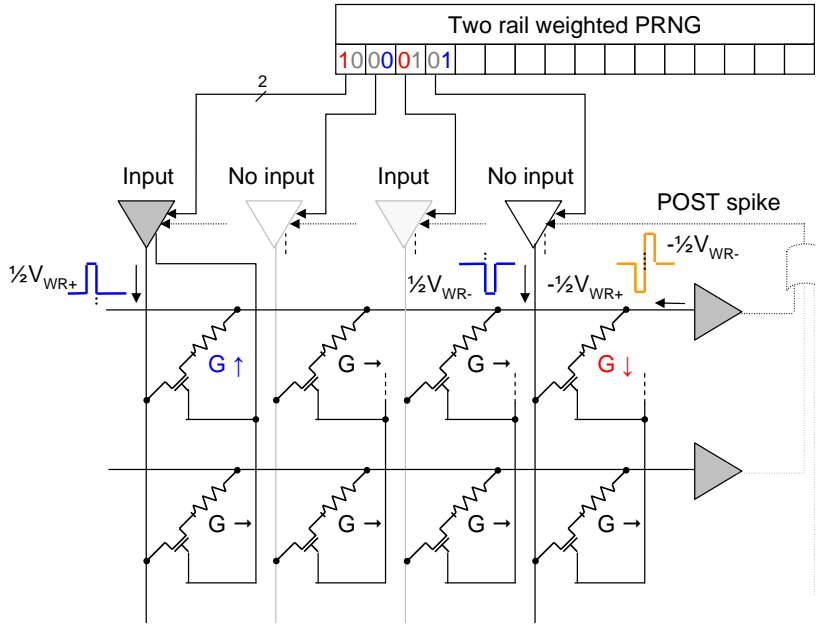


FIGURE 5.31 – Principe du système de programmation stochastique. De gauche à droite au niveau des neurones d’entrée : la LTP entraîne la commutation de la synapse ; la LTD n’entraîne pas la commutation de la synapse ; la LTP n’entraîne pas la commutation de la synapse ; la LTD entraîne la commutation de la synapse.

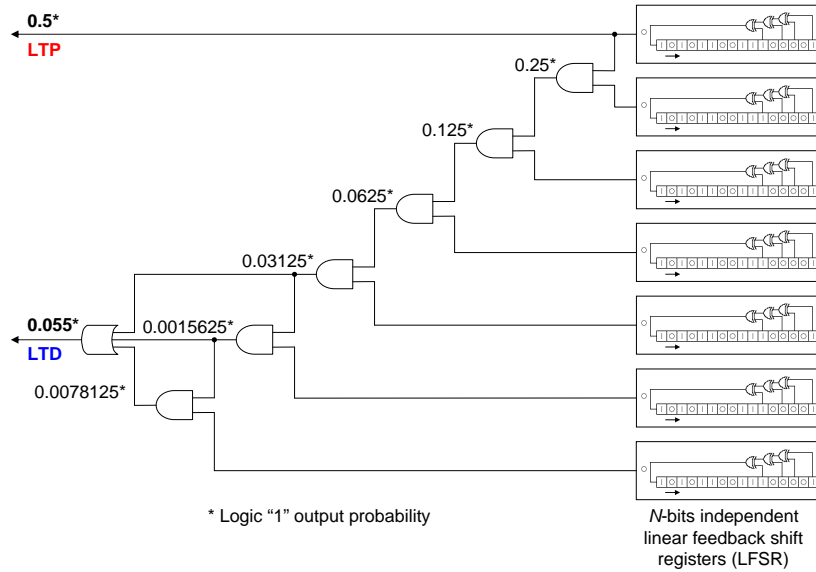


FIGURE 5.32 – Exemple d’implémentation d’un générateur pseudo-aléatoire biaisé pour implémenter les probabilités de LTP et LTD. Les probabilités sont obtenues en combinant les sorties de plusieurs LFSRs indépendants.

liste. Dans le cas de la CBRAM, ce circuit peut en outre tirer profit de la probabilité intrinsèque du dispositif. Les probabilités de commutation des dispositifs sont de nature physique et dépendent des conditions de programmation (amplitude et durée de l’impulsion). Elles sont indépendantes des probabilités de LTP et de LTD des PRNGs. La probabilité effective P de commutation du dispositif est donc le produit de la probabilité intrinsèque P_i et de la probabilité extrinsèque P_e :

$$P_{LTP/LTD} = P_{iLTP/LTD} \cdot P_{eLTP/LTD} \quad (5.7)$$

Il est donc possible de régler la probabilité extrinsèque $P_{eLTP/LTD}$ en fonction de $P_{iLTP/LTD}$ pour obtenir la probabilité $P_{LTP/LTD}$ désirée pour la règle d'apprentissage STDP (figure 5.30), qui est dépendante de l'application. Cette solution permet d'affaiblir les conditions de programmation des dispositifs pour optimiser la consommation énergétique du système. En effet, les impulsions de programmation normalement utilisées pour de la mémoire numérique nécessitent une amplitude et une durée suffisamment élevées pour garantir $P_i \approx 1.0$. Celles-ci peuvent potentiellement être significativement réduites même pour garder $P_i > 0.9$. Si les conditions de programmation sont choisies pour obtenir $P_{LTP/LTD} = P_{iLTP/LTD}$, les PRNGs peuvent être supprimés, économisant ainsi de l'énergie et de la surface silicium. Cette dernière solution est très élégante, mais elle suppose de pouvoir bien maîtriser la stochasticité des dispositifs avec les conditions de programmation, sans que ceux-ci ne se bloquent définitivement dans un état ON ou OFF après un certain nombre de commutations. Les trois modes de fonctionnement possibles, stochasticité extrinsèque, stochasticité intrinsèque et combinaison des deux, sont résumés sur la figure 5.33.

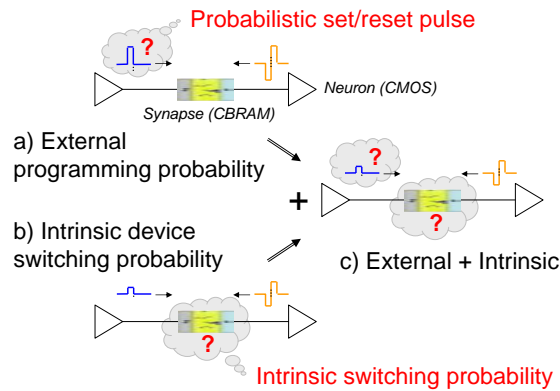


FIGURE 5.33 – Modes de fonctionnement : stochasticité extrinsèque, stochasticité intrinsèque et combinaison des deux.

5.5.2 Résultats de simulation

Afin de démontrer le potentiel de notre approche, nous avons simulé les trois applications présentées dans le chapitre «Étude d'architectures neuro-inspirées nouvelle génération pour le traitement de données naturelles» avec des synapses constituées d'un ou plusieurs dispositifs binaires avec de l'apprentissage STDP stochastique.

Extraction de trajectoires

L'application d'extraction de trajectoires de véhicules a été simulée avec une topologie identique à celle présentée dans la partie 4.2.3. De manière remarquable, un seul dispositif CBRAM par synapse, utilisant le modèle de la figure 5.29, permet d'extraire les trajectoires et de détecter les véhicules avec un taux supérieur à 95% sur les voies de circulation apprises. Les cartes de sensibilité des neurones de sortie, de la seconde couche, sont représentées sur la figure 5.34. Les paramètres utilisés pour les neurones et pour les probabilités P_{LTP} et P_{LTD} de la règle STDP stochastique sont donnés au tableau 5.4.

La consommation énergétique synaptique durant l'apprentissage est estimée à 74.2 μW .

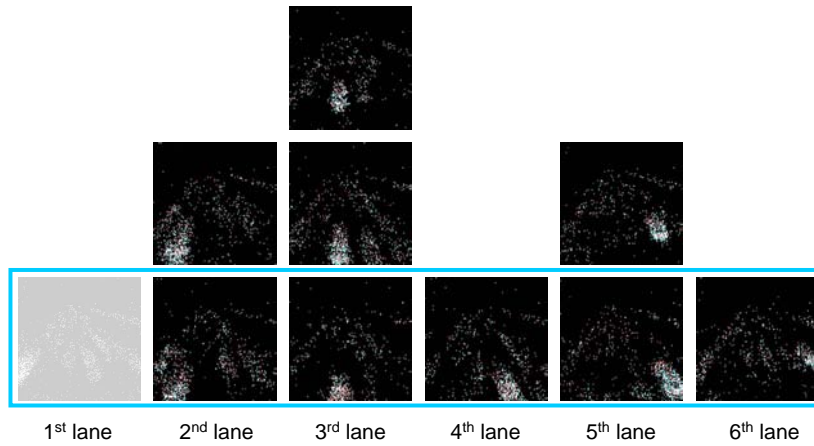


FIGURE 5.34 – Détection des véhicules sur les 6 voies de circulation par les 10 neurones de sortie de la seconde couche avec des synapses binaires. Chaque bloc montre la carte de sensibilité pour un neurone de sortie.

TABLE 5.4 – Paramètres pour l’apprentissage stochastique de trajectoires de véhicules. La stratégie d’apprentissage couche par couche est utilisée.

Paramètre	1ère couche	2ème couche
I_{thres}	0.201	0.000248
T_{LTP}	14.4 ms	1.84 ms
T_{refrac}	281 ms	530 ms
$T_{inhibit}$	43.8 ms	82.4 ms
τ_{leak}	80.2 ms	2.5 s
CBRAM/synapse	1	1
P_{LTP}	0.0905	0.612
P_{LTD}	0.153	0.138
Taux recon.	> 95% /voie apprise	

Extraction de motifs audio

Le protocole expérimental pour l’extraction de motifs audio a été présenté au chapitre précédent. Les caractéristiques de la séquence audio utilisée pour l’apprentissage, contenant le motif répétitif, ainsi que l’activité du neurone correspondant, sont présentées sur la figure 5.35. La séquence sonore est transcodée en trains d’impulsions en utilisant notre modèle simplifié de cochlée de 64 canaux. Ce transcodage peut également être réalisé matériellement avec une cochlée artificielle en CMOS utilisant le codage AER (Liu et al., 2010). Le réseau est constitué d’un seul neurone et de 3 CBRAM par synapse, soit en tout 192 dispositifs CBRAM, à raison d’une synapse connectant chaque canal au neurone.

La séquence d’apprentissage est initialement constituée de bruit blanc, de 0 à 400 s, et le neurone maintient une activité de sortie aléatoire. Ce résultat montre qu’une règle d’évolution des poids stochastique permet de maintenir une activité moyenne du neurone, ce qui n’est pas le cas d’une règle d’évolution additive avec des synapses graduelles, comme nous l’avons formalisé dans la dernière partie du chapitre précédent. Cette règle a donc la propriété de conserver un poids synaptique moyen intermédiaire stable. De 400 à 600 s, le motif constitué de bruit blanc est répété à des instants aléatoires dans la séquence de bruit. Le neurone devient rapidement sélectif au morceau répétitif et ne se déclenche plus sur du bruit après l’apprentissage, de 600 à 800 s.

La sélectivité du système a été évaluée de la même manière que dans le chapitre

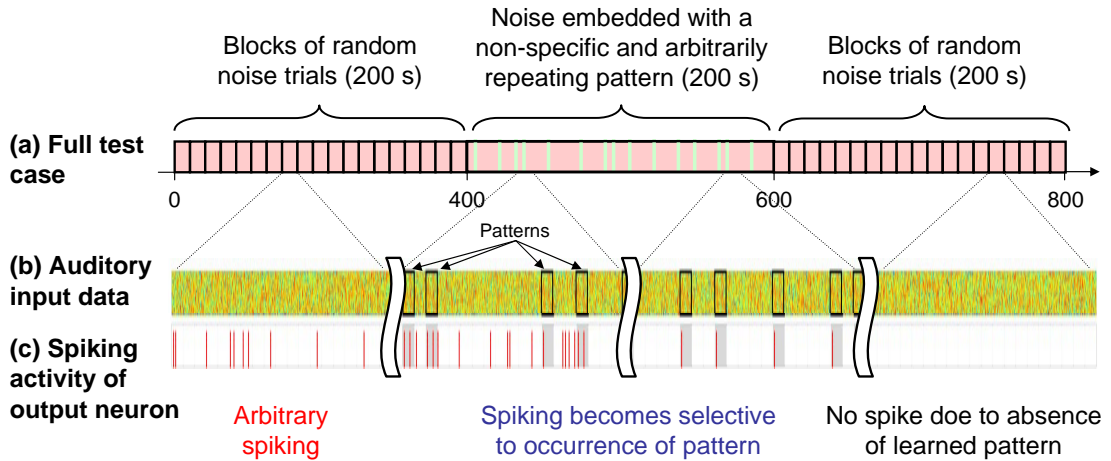


FIGURE 5.35 – Apprentissage d’un motif répétitif caché dans du bruit audio. (a) Description de la séquence de test pour l’apprentissage de données audio. (b) Spectrogramme de son en entrée. (c) Activité du neurone de sortie.

précédent et les résultats sont résumés sur la figure 5.36. Malgré l’utilisation de dispositifs purement binaires, la sélectivité est comparable à celle obtenue avec des synapses analogiques dans le chapitre précédent. Elle atteint un niveau similaire, voire supérieur, à celle de l’oreille humaine (Agus et al., 2010). Les paramètres utilisés pour les neurones et pour les probabilités P_{LTP} et P_{LTD} de la règle STDP stochastique sont donnés au tableau 5.5.

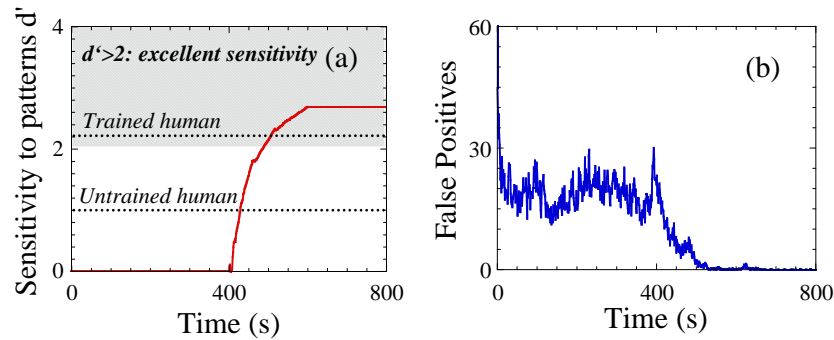


FIGURE 5.36 – Évolution de la sensibilité du neurone durant l’apprentissage. (a) Mesure de la sensibilité au motif répétitif du stimulus. (b) Mesure de l’activité du neurone pour 20 s de bruit blanc au fur et à mesure de la présentation de tranches de 1 s du stimulus.

La consommation énergétique synaptique du système est évaluée à $0.55 \mu\text{W}$.

TABLE 5.5 – Paramètres pour l’apprentissage stochastique de motifs auditifs.

Paramètre	Valeur
I_{thres}	0.00293
T_{LTP}	6.59 ms
T_{refrac}	38.8 ms
$T_{inhibit}$	N.A. (un seul neurone)
τ_{leak}	4.67 ms
CBRAM/synapse	3
P_{LTP}	0.514
P_{LTD}	0.0546

Classification de motifs

La classification de motifs a également été évaluée avec de l'apprentissage STDP stochastique. Sur la base MNIST, une redondance de 7 dispositifs binaires par synapse permet d'atteindre un taux de reconnaissance pratiquement identique à un apprentissage équivalent sur des synapses analogiques. Il est remarquable que même avec des synapses binaires (redondance de 1), le réseau est capable d'apprendre les motifs principaux constituant les chiffres manuscrits (figure 5.37). L'évolution du taux de reconnaissance en fonction de la redondance, comparé à des synapses analogiques, est représenté sur la figure 5.38. A noter que ces travaux, publiés dans (Querlioz et al., 2012b), utilisent un modèle générique de dispositifs intrinsèquement stochastiques et non pas le modèle de CBRAM comme ceci est le cas dans les deux exemples précédents.



FIGURE 5.37 – Reconstruction des poids pour un apprentissage stochastique sur la base MNIST.

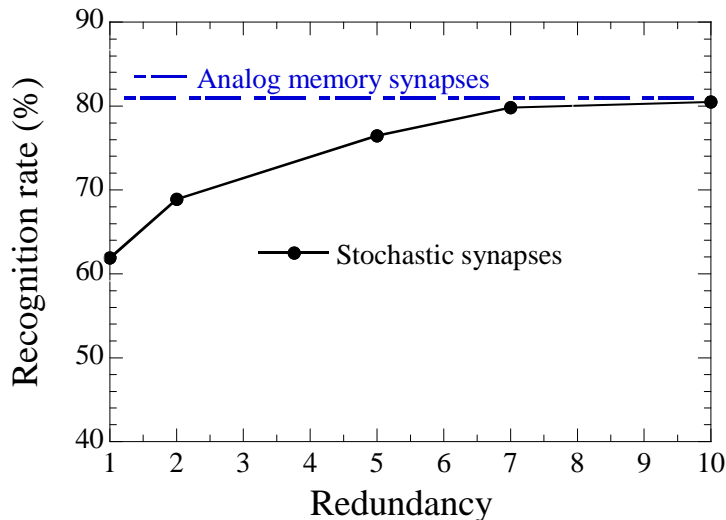


FIGURE 5.38 – Comparaison du taux de reconnaissance entre des synapses analogiques et des synapses binaires, en fonction de la redondance (nombre de dispositifs binaires par synapse).

5.6 Discussion et perspectives

Dans ce chapitre, nous avons proposé des implémentations d'apprentissage de type STDP sur des réseaux de neurones impulsifs sur trois dispositifs très différents : le NOMFET, la PCM et la CBRAM. L'une exploite la volatilité du NOMFET, l'autre la cumulativité du processus de cristallisation de la PCM et la dernière la stochasticité de la CBRAM. Ces travaux donnent un aperçu de l'étendue des possibilités offertes par la vaste famille des technologies émergentes de dispositifs mémoire memristifs, combinés à des architectures neuro-inspirées de nouvelle génération pour le traitement de données

naturelles. Nous espérons avoir établi dans ce chapitre des bases solides pour réaliser de telles architectures, qui ne demandent qu'à être développées.

De nombreux défis restent encore à relever, à la fois sur le plan technologique et sur le plan applicatif. Nous citerons notamment les problématiques liées à l'intégration en «crossbar» (courants de fuite, R_{on} et ratio R_{off}/R_{on} des dispositifs), à la réalisation pratique des neurones en CMOS (analogique/numérique, synchrone/asynchrone), à l'organisation topologique du réseau (taille des cellules, interconnexions)... etc.

Concrètement, la prochaine étape essentielle pour ces implémentations matérielles sur des technologies memristives est la réalisation d'un prototype intégré hybride nano/CMOS. Si de nombreuses tentatives ont eu lieu ou sont en cours et que des démonstrations à petites échelles ont déjà été réalisées, telles que pour le NOMFET, aucun circuit intégré comprenant une matrice de plusieurs milliers de dispositifs et réalisant une application complète n'a été réalisé à ce jour. L'impact d'un tel circuit, s'il remplit ses promesses, pourrait être absolument considérable dans la communauté et concrétiserait des idées développées depuis maintenant plus de quatre ans.

Chapitre 6

Conclusion, réflexions et perspectives

Sommaire

6.1 Synthèse	143
6.2 Réflexions et questions ouvertes	144
6.2.1 Les grands projets en cours	144
6.2.2 Quelles limitations pour STDP ?	147
6.2.3 Et l'implémentation des neurones ?	148
6.3 Perspectives	149

6.1 Synthèse

Dans cette thèse, nous avons exploré la possibilité de réaliser des architectures de traitement de données naturelles véritablement en ruptures avec les systèmes classiques basés sur des «Central Processing Unit»s (CPUs) ou des processeurs de signal numérique – «Digital Signal Processors» (DSPs). Nous avons montré comment l'on pouvait utiliser la memristivité de certaines technologies mémoire émergentes pour implémenter des synapses artificielles. Ces implémentations sont radicalement différentes de celles envisageables en technologie «Complementary Metal Oxide Semiconductor» (CMOS), qui se heurtent soit au goulot d'étranglement de Von Neumann, soit à une limitation importante de la densité d'intégration et de l'efficacité énergétique. Cela du fait du nombre de transistors et de la complexité du circuit requis pour émuler une synapse.

Grâce à une approche pluridisciplinaire, nous avons pu proposer des applications concrètes exploitant ces synapses. Celles-ci vont au-delà du simple mimétisme biologique, qui caractérise la majorité des travaux de l'état de l'art concomitant. Cela inclut une compréhension claire des mécanismes de «Spike-Timing-Dependent Plasticity» (STDP) pour l'extraction de motifs temporels et la classification de motifs, que nous avons partiellement formalisés.

Des analyses systématiques pour évaluer la faisabilité des architectures neuro-inspirées que nous proposons ont été entreprises. Celles-ci concernent en premier lieu le niveau de tolérance aux variabilités et aux défauts, inhérent aux dispositifs memristifs à l'échelle nano-métrique. Nous avons ainsi montré l'extrême robustesse des applications basées sur STDP face à ces défauts. La tolérance au bruit et l'efficacité énergétique ont également été évalués.

Nous pensons, à l'issue de ces travaux, avoir répondu de manière convaincante à

l'enjeu majeur évoqué dans l'introduction. A savoir, de déterminer la faisabilité d'applications concrètes et robustes basées sur des technologies memristives.

6.2 Réflexions et questions ouvertes

Dans cette partie, je souhaite engager quelques réflexions de fond sur mes travaux et plus généralement sur le domaine des nouvelles architectures à base de dispositifs memristifs. Je souhaite également partager mon opinion, qu'elle soit argumentée ou non, sur certaines questions essentielles.

6.2.1 Les grands projets en cours

SyNAPSE («Systems of Neuromorphic Adaptive Plastic Scalable Electronics»), financé par la Defense Advanced Research Projects Agency (DARPA). Ce projet très ambitieux, lancé en 2008, vise à développer de la technologie neuromorphique capable d'atteindre une densité d'intégration comparable à la biologie. L'effort dans ce projet porte principalement sur l'intégration de technologies memristives pour implémenter des synapses et des fonctionnalités telles que STDP. Le plus gros partenaire du projet est IBM, sous la direction de Dharmendra Modha (Modha and Shenoy, 2010; Breitwisch et al., 2010; Jackson et al., 2011). Le second partenaire principal est HRL Laboratories. Parmi les autres partenaires ou anciens partenaires, on retrouve entre autres, l'Université de Stanford, avec H.-S. Philip Wong (Yu and Wong, 2010; Liang and Wong, 2010; Liang et al., 2011; Kuzum et al., 2012), l'Université du Michigan, avec Wei Lu (Jo and Lu, 2008; Jo et al., 2009a,b, 2010; Kim et al., 2012) et l'Université de Boston, avec Stephen Grossberg, pour la partie applicative. La figure 6.1 résume l'objectif du programme et la démarche adoptée.

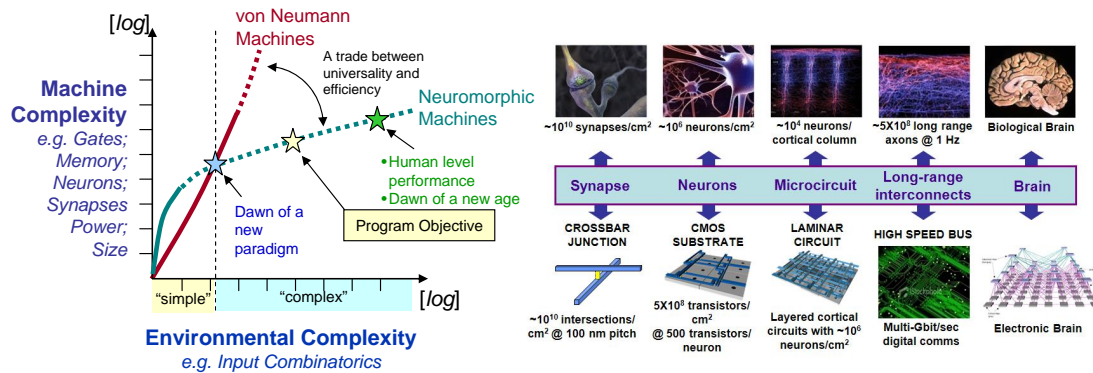


FIGURE 6.1 – Vision du projet SyNAPSE de la DARPA (images publiques du projet). A gauche, l'objectif du programme, mettant face à face les architectures Von Neumann et les architectures neuromorphiques. A droite, la démarche adoptée pour atteindre une densité d'intégration comparable à la biologie.

A ce jour, l'objectif initial du projet, qui visait à intégrer l'équivalent de la capacité cognitive du cerveau d'un chat dans un circuit intégré, est semble-t-il, sans grande surprise, loin d'être atteint. Le projet a néanmoins beaucoup fait parler de lui avec la simulation, sur un supercalculateur, d'un réseau de neurones avec le même nombre de synapses et de neurones que dans le cerveau d'un chat (Ananthanarayanan et al., 2009). Ce succès est à relativiser, car il s'agit avant tout d'un résultat technique sur la simulation à grande échelle d'un réseau avec des modèles simples. Celui-ci ne réalise malheureusement aucune fonction cognitive (mis à part afficher le logo d'IBM en trains d'impulsions).

La vision de l'instigateur du projet, Todd Hylton, est que l'intelligence devrait émerger naturellement à partir d'un réseau suffisamment complexe et s'appuie pour cela sur des arguments de thermodynamique. Scientifiquement, ce type d'argumentation est intéressant, mais aussi discutable. Je pense que la compréhension des mécanismes d'apprentissage est au contraire essentielle pour progresser à partir de réseaux élémentaires vers des structures plus complexes.

D'autres développements dans le cadre de ce projet ont reçu une large publicité. On peut citer MoNETA («MOdular Neural Exploring Traveling Agent»), développé à l'Université de Boston (Ames et al., 2012), qui s'appuie sur le logiciel Cog Ex Machina, développé par Greg Snider (Snider et al., 2011). Le premier permet de modéliser un agent complet capable de se déplacer dans l'environnement tandis que le second offre une interface avec un supercalculateur ou du matériel neuromorphique à base de dispositifs memristifs.

Ces outils ont permis pour le moment de réaliser un agent virtuel capable de résoudre le Labyrinthe de Morris. Dans cette expérience, l'agent (un rat ou une souris dans la réalité), est déposé dans un piscine qui contient une petite plateforme immergée à quelques millimètres de la surface. L'agent doit apprendre à trouver cette plateforme pour se maintenir hors de l'eau. Pour cela, des indices visuels sont placés sur le bord de la piscine (ronds de différentes couleurs par exemple). Le système mis en œuvre pour implémenter cette tâche n'a malheureusement pas encore été présenté en détails. Il est donc difficile d'apprécier la pertinence de ces travaux, présentés ni plus ni moins comme l'avènement de l'intelligence artificielle forte, après 50 ans de promesses non tenues (Versace and Chandler, 2010)... L'histoire se répèterait-elle ?

BrainScales¹ («Brain-inspired multiscale computation in neuromorphic hybrid systems») est un projet européen «Future and emerging technologies» (FET) FP7, démarré en 2011 et qui fait suite au projet FACETS («Fast Analog Computing with Emergent Transient States», 2005-2010). Karlheinz Meier, de l'Université de Heidelberg, en est le coordinateur. Le projet vise à fabriquer du matériel neuromorphique à l'échelle «multi-wafer» dans le but d'accélérer des simulations de neuroscience. Celles-ci concernent notamment l'analyse d'expérimentations biologiques in-vivo, pour la compréhension des fonctions et des interactions à plusieurs échelles temporelles et spatiales pour le traitement de l'information dans le cerveau. Temporellement, les phénomènes de plasticité peuvent aller de la milliseconde à des heures ou à des jours. Spatialement, les interactions ont lieu au niveau des neurones individuels, de populations de neurones, jusqu'à des zones entières du cerveau. La figure 6.2 montre une représentation 3D d'un module matériel à l'échelle d'un «wafer» destiné à être réalisé dans le projet.

L'objectif de ce projet est d'accélérer les simulations biologiques. Un facteur d'accélération pouvant aller jusqu'à 10 000 par rapport au temps biologique est projeté. Il utilise pour cela de la technologie CMOS classique et propose une implémentation principalement analogique des neurones et de la plasticité synaptique (Schemmel et al., 2010). La technologie mise en œuvre dans ce projet est particulièrement impressionnante. Cependant, l'inconvénient principal d'une telle implémentation est le manque de flexibilité. Cela est à mon avis d'autant plus vrai pour un système qui se destine essentiellement à des études de neurosciences. Les modèles de neurones et de plasticité sont figés. Ils disposent certes d'un certain nombre de paramètres réglables, mais au prix d'une complexité importante. En outre, les paramètres pris en compte, par rapport aux différents modèles biologiques, découlent d'un choix qui reste en partie arbitraire, et qui est subordonné par la complexité de leur implémentation.

La motivation principale pour créer du matériel neuromorphique dans ce projet est

1. Site Internet : <http://brainscales.kip.uni-heidelberg.de/>

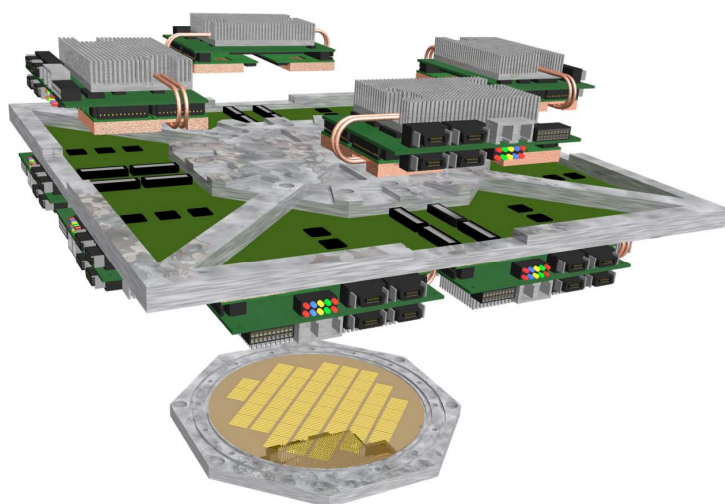


FIGURE 6.2 – Vision du projet européen BrainScaleS (image publique du projet). Il s’agit d’un module matériel de réseau de neurones, intégré directement à l’échelle d’un «wafer».

la même que celle introduite dans les années 1990. Il s’agit d’implémenter matériellement des architectures qui seraient trop lentes sur des CPU classiques, à cause du goulot d’étranglement de Von Neumann. Cependant, j’ai défendu tout au long de cette thèse qu’une implémentation purement en CMOS reste inefficace pour résoudre ce problème. Par ailleurs, contrairement au cas d’un système embarqué, l’efficacité n’est ici pas une motivation.

Pour ces raisons, je reste persuadé que les objectifs affichés du projet pour la simulation pourraient être réalisés avec des supercalculateurs, qui auraient l’avantage d’offrir bien plus de flexibilité et d’évolutivité. C’est d’ailleurs cette dernière approche qui est choisie dans le projet Blue Brain. BrainScaleS développe cependant une technologie d’intégration à l’échelle «wafer» qui pourrait bénéficier à de nombreux type de projets dans le futur.

SpiNNaker² (contraction pour «Spiking Neural Network Architecture») est un projet d’architecture neuromorphique massivement parallèle basée sur des cœurs ARM9 (Furber et al., 2006). Il est mené par l’Université de Manchester. Contrairement au projet BrainScaleS, les neurones et les synapses sont entièrement programmables. Un cœur ARM9 peut ainsi émuler de quelques neurones complexes utilisant des modèles multi-compartimentaux à des centaines de neurones simples de type «Leaky integrate-and-fire» (LIF). Le projet vise à créer une architecture en réseau de plus d’un million de cœurs ARM9. Chaque nœud du réseau, constituant un «System In Package» (SIP), comporte 18 cœurs.

La finalité du projet est technique, ARM étant un partenaire important, et se veut constituer une plateforme pour l’implémentation de réseaux de neurones assez génériques. Je situerais donc l’approche du projet entre BrainScaleS et Blue Brain, en termes de spécialisation de l’implémentation. Malgré des innovations sur la communication et la transmission des événements entre les nœuds d’un réseau, SpiNNaker se heurte également au «bottleneck» de Von Neumann. Cela est particulièrement vrai pour l’implémentation de STDP, qui n’est pas simple avec les ressources limitées des cœurs de calcul (Jin et al., 2010).

En étant plus flexible que BrainScaleS et plus adaptée qu’un supercalculateur classique pour le neuromorphique, je pense que l’approche adoptée a du sens pour des si-

2. Site Internet : <http://apt.cs.manchester.ac.uk/projects/SpiNNaker/>

mulations de neurosciences. L'intérêt par rapport à un supercalculateur n'est toutefois pas évident. Les objectifs scientifiques du projet ne sont pas à mon sens extrêmement clairs. Si l'on se cantonne à de la simulation pour les neurosciences, l'intérêt d'avoir un système plus efficace énergétiquement est essentiellement économique (notamment par rapport à l'investissement conséquent que représente l'exploitation d'un supercalculateur). Mais cela risque de ne plus être vrai après quelques années grâce à la loi de Moore.

Les autres projets. Outre ces trois projets, on citera également Neurogrid³, de l'Université de Stanford, dirigé par Kwabena Boahen et Blue Brain, le projet de Henry Markram, à l'École Polytechnique Fédérale de Lausanne (EPFL). L'approche de Neurogrid est similaire à celle de BrainScaleS, à plus petite échelle. Le projet Blue Brain utilise quant à lui des supercalculateurs classiques pour faire des simulations détaillées du cerveau. Il vise à terme à recréer un cerveau virtuel. Par rapport aux autres projets, le projet cherche à pousser le niveau de réalisme des modèles le plus loin possible pour reproduire parfaitement les processus biologiques dans le cerveau.

Enfin, le Human Brain Project (HBP) est une proposition de projet de grande envergure, qui aspire à devenir un projet «flagship» de l'Union Européenne dès 2013. Celui-ci ambitionne de munir l'Europe d'une structure de recherche dédiée à la simulation du cerveau humain. S'il est retenu, l'EPFL coordonnera ce projet, qui s'appuiera sur les travaux déjà menés dans le cadre des projets Blue Brain et BrainScaleS.

6.2.2 Quelles limitations pour STDP ?

Mes travaux traitent de plusieurs applications de traitement de données sur des réseaux de neurones impulsionsnels, qui sont toutes basées sur un algorithme d'apprentissage de type STDP. C'est aujourd'hui le cas de la grande majorité des travaux sur les réseaux de neurones impulsionsnels dans la communauté du neuromorphique. Ceux-ci considèrent toutefois parfois qu'émuler l'une des quelques mesures biologiques disponibles pour STDP permettra de réaliser ensuite des applications «cognitives». Parfois avec quelques raccourcis. Je pense à l'aspect vraiment cumulatif de STDP qui est rarement démontré à la fois pour la potentialisation à long terme – «Long-Term Potentiation» (LTP) et la dépression à long terme – «Long-Term Depression» (LTD) (on a vu que ce n'était d'ailleurs pas le cas ni pour la mémoire à changement de phase – «Phase-Change Memory» (PCM) ni la «Conductive-Bridging RAM» (CBRAM)). Cela montre à mon avis qu'il est pour le moins prématuré de proposer des implémentations de STDP sans avoir au préalable exploré les applications qu'il est possible de faire avec. Et, à la lumière de ces travaux de thèse, que le modèle classique «bio-réaliste» n'est peut-être pas le plus prometteur.

Au-delà du modèle en lui-même et de son implémentation, STDP comporte tout de même certaines interrogations : qu'en est-il du passage à l'échelle, de l'apport du multi-couches, des problèmes non linéairement séparables, de l'apport des poids négatifs... ? L'apport du multi-couches en particulier est une question qui me semble particulièrement intéressante. Je ne parviens pas pour le moment à concevoir un système de plusieurs couches qui améliorerait les performances d'un apprentissage d'une manière aussi importante qu'un perceptron multi-couches – «Multilayer Perceptron» (MLP). L'intérêt de poids négatifs, ou inhibiteurs, n'est pas non plus totalement élucidé, mais pourrait être l'une des clés pour le multi-couches. Un simple perceptron est incapable de résoudre un problème non linéairement séparable. La question se pose également pour

3. Site Internet : <http://www.stanford.edu/group/brainsinsilicon/>

une couche STDP. Mais qu'est-ce qu'un problème non linéairement séparable avec un codage temporel de l'information et STDP ?

La résolution de ces problématiques est pour moi une priorité, sans quoi STDP pourrait ne pas délivrer ses promesses. Mais pour le moment je ne suis pas inquiet, il reste beaucoup de choses à explorer.

6.2.3 Et l'implémentation des neurones ?

Dans ces travaux, nous avons avant tout traité de la manière d'exploiter des dispositifs memristifs en tant que synapses. Nous avons simplement supposé que les neurones sont implémentés en technologie CMOS. En CMOS, c'est clairement la taille de la synapse qui limite la densité d'intégration et l'efficacité énergétique, à raison de 1000 à 10 000 synapses par neurone, si l'on retient l'ordre de grandeur biologique. Dans le cas d'un système hybride implémentant les synapses avec des dispositifs memristifs, ce pourrait bien être au contraire le neurone qui est le facteur limitant.

Cette considération peut être lourde de conséquences, alors que les neurones occupent déjà un aspect central dans toute architecture neuro-inspirée et que leur implémentation n'est pas triviale. La meilleure manière d'implémenter le neurone est donc le sujet de nombreux débats. L'une des questions importantes est de savoir s'il est préférable de l'implémenter en analogique ou en numérique (Joubert et al., 2012). Cela est plus vrai encore dans une implémentation hybride nano/CMOS, qui implique forcément une interface analogique entre le neurone et ses synapses.

Il n'existe sans doute pas de réponse universelle à cette question, car tout dépend du modèle du neurone, de l'apprentissage et de l'architecture pour la programmation des synapses. Je pense qu'une implémentation essentiellement numérique a l'énorme avantage de la flexibilité et de la programmabilité. Nous avons vu dans ces travaux que les paramètres des neurones, tels que les constantes de temps, pour l'inhibition, la LTP, la fuite, étaient fortement liés à la dynamique des stimuli à apprendre. Ceux-là pourraient être stockés dans des registres programmables, par couches de neurones par exemple. Une implémentation analogique est également beaucoup plus délicate à concevoir et est soumise à la variabilité des transistors CMOS.

Au niveau de l'efficacité énergétique, la nécessité de distribuer une horloge dans le circuit dans le cas d'une implémentation numérique n'est pas selon moi un facteur limitant. La résolution temporelle minimale pour tenir compte de la dynamique d'apprentissage dans nos applications ne dépasse pas l'ordre de la milliseconde (c'est la résolution couramment utilisée dans les simulations de réseaux biologiques en neurosciences). Ainsi, une fréquence d'horloge numérique inférieure au mégahertz pourrait être suffisante pour faire fonctionner un système en temps réel, bien plus faible en tout cas que les fréquences utilisées dans les processeurs actuels. En outre, dans un système à grande échelle, distribué sur plusieurs puces, l'énergie nécessaire à la communication entre les neurones pourrait dominer celle du neurone lui-même.

Un bon compromis pourrait donc être de conserver la sommation analogique des courants en sortie du «crossbar» et de l'intégrer dans un condensateur qui serait échantillonné toutes les millisecondes ou 0.1 ms avec un convertisseur analogique numérique (CAN). Une période d'échantillonnage relativement faible permet d'implémenter le terme de fuite numériquement et de réduire au minimum la taille du condensateur d'intégration. La taille de ce dernier peut être considérable et fortement limiter la densité d'intégration des neurones.

6.3 Perspectives

Avant de refermer ce mémoire, j'aimerais exposer, très brièvement, les directions que je souhaite suivre dans le futur. Comme cela a déjà été le cas tout au long de cette thèse, je pense qu'il est indispensable de continuer à avancer de manière synchronisée sur les différents fronts :

- **Au niveau applicatif**, s'efforcer de répondre aux interrogations évoquées précédemment sur STDP. Explorer de nouvelles applications et étendre celles déjà étudiées. Dans le cas de la détection de trajectoires de véhicules, on peut envisager de rajouter une étape de classification, portant sur le type ou le modèle de véhicule par exemple.
- **Au niveau technologique**, continuer les caractérisations des dispositifs memristifs. Après la PCM et la CBRAM, la prochaine étape au Laboratoire d'électronique des technologies de l'information du Commissariat à l'énergie atomique et aux énergies alternatives (CEA) (CEA LETI) est de caractériser des technologies mémoire résistive – «Resistive RAM» (RRAM). Il paraît important également de chercher à optimiser les matériaux et procédés de fabrication pour une utilisation synaptique des dispositifs.
- **Au niveau circuit**, élargir les réflexions sur l'implémentation des neurones et l'intégration en «crossbar». Ces deux points restent critiques pour le développement d'un circuit hybride nano/CMOS complet.

Sur une vision un peu plus long terme, même au regard des résultats prometteurs présentés dans ce mémoire, il est légitime de réfléchir au devenir de ce neuromorphique nouvelle génération. Surtout si l'on considère l'historique des réseaux de neurones, qui ne joue pas nécessairement en leur faveur. Aujourd'hui, parler de réseaux de neurones à certains industriels est toujours quelque peu tabou...

Nous avons toutefois évoqué de nombreuses raisons pour tenter d'expliquer cet état de fait et aussi pourquoi la donne devrait changer dans le futur grâce aux solutions que nous proposons. Je pense honnêtement que le diagnostic et la solution proposée vont dans le bon sens. La question est malgré tout de savoir si cela sera suffisant, passé l'effet de mode inspirée par le memristor et les réseaux de neurones impulsifs, pour venir justifier l'inclusion de matériel neuromorphique spécialisé dans des produits de masse. Ou si la loi de Moore continuera à avoir le dernier mot.

Personnellement, je ne suis pas certain que cette révolution du neuromorphique, attendue depuis des décennies, ait lieu immédiatement. Peut-être même pas dans quelques années. Il y a encore je pense de nombreuses choses à découvrir au préalable. Mais les progrès dans ce domaine sont constants, malgré des hauts et des bas dans l'histoire encore balbutiante du neuromorphique. Je suis donc à peu près certain que cette révolution finira par se produire et que les neurosciences joueront un rôle central dans cet accomplissement. Et cela pour une raison simple : l'évolution n'a pas retenue le codage binaire et la logique booléenne pour créer des êtres intelligents et adaptés à leur environnement. Et je ne pense pas que cela était hors de portée pour la nature.

Glossaire

AER «Address-Event Representation».

AMS mixte analogique/numérique – «Analog and Mixed-Signal».

AMTL Advanced Memory Technology Laboratory.

CAM mémoire adressable par contenu – «Content-Addressable Memory».

CAN convertisseur analogique numérique.

CAO conception assistée par ordinateur.

CBRAM «Conductive-Bridging RAM».

CEA Commissariat à l'énergie atomique et aux énergies alternatives.

CEA LETI Laboratoire d'électronique des technologies de l'information du CEA.

CEA LIST Laboratoire d'Intégration des Systèmes et Technologies du CEA.

CMOL «CMOS/molecular».

CMOS «Complementary Metal Oxide Semiconductor».

CNRS Centre national de la recherche scientifique.

CPU «Central Processing Unit».

CRS «Complementary Resistive Switches».

DACLE Département Architectures Conception & Logiciels Embarqués.

DARPA Defense Advanced Research Projects Agency.

DSP processeur de signal numérique – «Digital Signal Processor».

DTSI Département Technologies et Systèmes Intelligents.

EPFL École Polytechnique Fédérale de Lausanne.

EPSC courant post-synaptique excitateur – «Excitatory Postsynaptic Current».

ERB bande passante rectangulaire équivalente – «Equivalent Rectangular Bandwidth».

FET «Future and emerging technologies».

FFT transformée de Fourier rapide – «Fast Fourier Transform».

FPGA «Field-Programmable Gate Array».

FPNI «Field-Programmable Nanowire Interconnect».

GPU «Graphics Processing Unit».

HBP Human Brain Project.

HMM modèle de Markov caché – «Hidden Markov Model».

HP Hewlett-Packard.

-
- HRS** «High Resistance State».
- IDC** International Data Corporation.
- IEF** Institut d'Électronique Fondamentale.
- IEMN** Institut d'Électronique, de Microélectronique et de Nanotechnologie.
- ISA** «Instruction Set Architecture».
- ITRS** International Technology Roadmap for Semiconductors.
- LCE** Laboratoire Calcul Embarqué.
- LFSE** Laboratoire Fiabilisation des Systèmes Embarqués.
- LFSR** registre à décalage à rétroaction linéaire – «Linear Feedback Shift Register».
- LIF** «Leaky integrate-and-fire».
- LRS** «Low Resistance State».
- LTD** dépression à long terme – «Long-Term Depression».
- LTP** potentialisation à long terme – «Long-Term Potentiation».
- MAC** multiplie puis accumule – «Multiply-Accumulate».
- MIM** métal-isolant-métal.
- MLP** perceptron multi-couches – «Multilayer Perceptron».
- MOS** «Metal Oxide Semiconductor».
- MRAM** mémoire magnétique – «Magnetic RAM».
- MTJ** jonction tunnel magnétique – «Magnetic Tunnel Junction».
- NOMFET** «Nanoparticle-Organic Memory Field Effect Transistor».
- NP** nano-particule.
- PCM** mémoire à changement de phase – «Phase-Change Memory».
- PMC** «Programmable-Metallization-Cell».
- PMMA** polyméthacrylate de méthyle – «Polymethyl Methacrylate».
- PRNG** générateur de nombres pseudo-aléatoires – «Pseudorandom Number Generator».
- ROI** «Region Of Interest».
- RRAM** mémoire résistive – «Resistive RAM».
- SARC** Service Architectures et Conception.
- SEM** microscopie électronique à balayage – «Scanning Electron Microscopy».
- SIMD** «Single Instruction Multiple Data».
- SIP** «System In Package».
- STDP** «Spike-Timing-Dependent Plasticity».
- STL** bibliothèque standard C++ – «Standard Template Library».
- TMR** magnétorésistance à effet tunnel – «Tunnel Magnetoresistance».
- TO** «Transistor Outline».
- WAV** «WAVEform audio file format».
- WTA** «Winner-Take-All».

Publications personnelles

Articles de revues internationales

- F. Alibart, S. Pleutin, O. Bichler, C. Gamrat, T. Serrano-Gotarredona, B. Linares-Barranco, and D. Vuillaume. A memristive nanoparticle/organic hybrid synapstor for neuroinspired computing. *Advanced Functional Materials*, 22(3):609–616, 2012. doi:[10.1002/adfm.201101935](https://doi.org/10.1002/adfm.201101935).
- O. Bichler, W. Zhao, F. Alibart, S. Pleutin, D. Vuillaume, and C. Gamrat. Functional model of a nanoparticle organic memory transistor for use as a spiking synapse. *Electron Devices, IEEE Transactions on*, 57(11):3115–3122, 2010. doi:[10.1109/TED.2010.2065951](https://doi.org/10.1109/TED.2010.2065951).
- O. Bichler, D. Querlioz, S. Thorpe, J. Bourgoïn, and C. Gamrat. Extraction of temporally correlated features from dynamic vision sensors with spike-timing-dependent plasticity. *Neural Networks*, 32:339–348, 2012a. doi:[10.1016/j.neunet.2012.02.022](https://doi.org/10.1016/j.neunet.2012.02.022).
- O. Bichler, M. Suri, D. Querlioz, D. Vuillaume, B. DeSalvo, and C. Gamrat. Visual pattern extraction using energy-efficient '2-PCM synapse' neuromorphic architecture. *Electron Devices, IEEE Transactions on*, 59(8):2206–2214, 2012b. doi:[10.1109/TED.2012.2197951](https://doi.org/10.1109/TED.2012.2197951).
- O. Bichler, W. Zhao, F. Alibart, S. Pleutin, D. Vuillaume, and C. Gamrat. Pavlov's dog associative learning demonstrated on synaptic-like organic transistors. *Neural Computation*, 2012c. doi:[10.1162/NECO_a_00377](https://doi.org/10.1162/NECO_a_00377). In press.
- D. Querlioz, O. Bichler, P. Dollfus, and C. Gamrat. Unsupervised neuro-inspired system with memristive devices, immune to device variations. *Neural Networks*, 2012a. Submitted.
- D. Querlioz, O. Bichler, W. S. Zhao, J.-O. Klein, P. Dollfus, and C. Gamrat. Stochastic nanodevices for a bio-inspired system capable of learning. *Applied Physics Letters*, 2012b. Submitted.
- M. Suri, O. Bichler, D. Querlioz, B. Traoré, O. Cueto, L. Perniola, V. Sousa, D. Vuillaume, C. Gamrat, and B. DeSalvo. Physical aspects of low power synapses based on phase change memory devices. *Journal of Applied Physics*, 112(5):054904, 2012. doi:[10.1063/1.4749411](https://doi.org/10.1063/1.4749411).

Actes de la conférence IEDM

- M. Suri, O. Bichler, D. Querlioz, O. Cueto, L. Perniola, V. Sousa, D. Vuillaume, C. Gamrat, and B. DeSalvo. Phase change memory as synapse for ultra-dense neuromorphic systems: Application to complex visual pattern extraction. In *Electron Devices Meeting (IEDM), 2011 IEEE International*, pages 4.4.1–4.4.4, 2011. doi:[10.1109/IEDM.2011.6131488](https://doi.org/10.1109/IEDM.2011.6131488).

M. Suri, O. Bichler, D. Querlioz, G. Palma, E. Vianello, D. Vuillaume, C. Gamrat, and B. DeSalvo. CBRAM devices as binary synapses for low-power stochastic neuromorphic systems: Auditory (cochlea) and visual (retina) cognitive processing applications. In *Electron Devices Meeting (IEDM), 2012 IEEE International*, 2012. In press.

Autres actes de conférences

O. Bichler, W. Zhao, C. Gamrat, F. Alibart, S. Pleutin, and D. Vuillaume. Development of a functional model for the nanoparticle-organic memory transistor. In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pages 1663–1666, 2010. doi:[10.1109/ISCAS.2010.5537487](https://doi.org/10.1109/ISCAS.2010.5537487).

O. Bichler, D. Querlioz, S. Thorpe, J. Bourgoïn, and C. Gamrat. Unsupervised features extraction from asynchronous silicon retina through spike-timing-dependent plasticity. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 859–866, 2011. doi:[10.1109/IJCNN.2011.6033311](https://doi.org/10.1109/IJCNN.2011.6033311).

D. Querlioz, O. Bichler, and C. Gamrat. Simulation of a memristor-based spiking neural network immune to device variations. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 1775–1781, 2011a. doi:[10.1109/IJCNN.2011.6033439](https://doi.org/10.1109/IJCNN.2011.6033439).

D. Querlioz, P. Dollfus, O. Bichler, and C. Gamrat. Learning with memristive devices: How should we model their behavior? In *Nanoscale Architectures (NANOARCH), 2011 IEEE/ACM International Symposium on*, pages 150–156, 2011b. doi:[10.1109/NANOARCH.2011.5941497](https://doi.org/10.1109/NANOARCH.2011.5941497).

D. Querlioz, W. Zhao, P. Dollfus, J.-O. Klein, O. Bichler, and C. Gamrat. Bioinspired networks with nanoscale memristive devices that combine the unsupervised and supervised learning approaches. In *Nanoscale Architectures (NANOARCH), 2012 IEEE/ACM International Symposium on*, 2012. In press.

M. Suri, O. Bichler, Q. Hubert, L. Perniola, V. Sousa, C. Jahan, D. Vuillaume, C. Gamrat, and B. DeSalvo. Interface engineering of PCM for improved synaptic performance in neuromorphic systems. In *Memory Workshop (IMW), 2012 4th IEEE International*, pages 1–4, 2012. doi:[10.1109/IMW.2012.6213674](https://doi.org/10.1109/IMW.2012.6213674).

Bibliographie

- L. F. Abbott, J. A. Varela, K. Sen, and S. B. Nelson. Synaptic depression and cortical gain control. *Science*, 275(5297):221–224, 1997. doi:[10.1126/science.275.5297.221](https://doi.org/10.1126/science.275.5297.221).
- Verilog-A Manual*. Agilent Technology, 2007. <http://www.designersguide.org/>.
- G. Agnus, W. Zhao, V. Derycke, A. Filoramo, Y. Lhuillier, S. Lenfant, D. Vuillaume, C. Gamrat, and J.-P. Bourgoïn. Two-terminal carbon nanotube programmable devices for adaptive architectures. *Advanced Materials*, 22(6):702–706, 2010. doi:[10.1002/adma.200902170](https://doi.org/10.1002/adma.200902170).
- T. R. Agus, S. J. Thorpe, and D. Pressnitzer. Rapid formation of robust auditory memories: Insights from noise. *Neuron*, 66(4):610–618, 2010. doi:[10.1016/j.neuron.2010.04.014](https://doi.org/10.1016/j.neuron.2010.04.014).
- F. Alibart, S. Pleutin, D. Guérin, C. Novembre, S. Lenfant, K. Lmimouni, C. Gamrat, and D. Vuillaume. An organic nanoparticle transistor behaving as a biological spiking synapse. *Advanced Functional Materials*, 20(2):330–337, 2010. doi:[10.1002/adfm.200901335](https://doi.org/10.1002/adfm.200901335).
- F. Alibart, T. Sherwood, and D. Strukov. Hybrid cmos/nanodevice circuits for high throughput pattern matching applications. In *Adaptive Hardware and Systems (AHS), 2011 NASA/ESA Conference on*, pages 279–286, 2011. doi:[10.1109/AHS.2011.5963948](https://doi.org/10.1109/AHS.2011.5963948).
- F. Alibart, L. Gao, B. D. Hoskins, and D. B. Strukov. High precision tuning of state for memristive devices by adaptable variation-tolerant algorithm. *Nanotechnology*, 23(7):075201, 2012a.
- F. Alibart, E. Zamanidoost, and D. B. Strukov. Pattern classification by memristive crossbar circuits with ex-situ and in-situ hebbian training. 2012b.
- H. Ames, E. Mingolla, A. Sohail, B. Chandler, A. Gorchetchnikov, J. Leveille, G. Livitz, and M. Versace. The animat: New frontiers in whole brain modeling. *Pulse, IEEE*, 3(1):47–50, 2012. doi:[10.1109/MPUL.2011.2175638](https://doi.org/10.1109/MPUL.2011.2175638).
- R. Ananthanarayanan, S. K. Esser, H. D. Simon, and D. S. Modha. The cat is out of the bag: cortical simulations with 10^9 neurons, 10^{13} synapses. In *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis, SC '09*, pages 63:1–63:12. ACM, 2009. doi:[10.1145/1654059.1654124](https://doi.org/10.1145/1654059.1654124).
- A. Anzai, X. Peng, and D. C. Van Essen. Neurons in monkey visual area V2 encode combinations of orientations. *Nature Neuroscience*, 10:1313–1321, 2007. doi:[10.1038/nm1975](https://doi.org/10.1038/nm1975).
- J. Backus. Can programming be liberated from the von neumann style?: a functional style and its algebra of programs. *Commun. ACM*, 21(8):613–641, 1978. doi:[10.1145/359576.359579](https://doi.org/10.1145/359576.359579).

- I. Baek, M. Lee, S. Seo, M. Lee, D. Seo, D.-S. Suh, J. Park, S. Park, H. Kim, I. Yoo, U.-I. Chung, and J. Moon. Highly scalable nonvolatile resistive memory using simple binary oxide driven by asymmetric unipolar voltage pulses. In *Electron Devices Meeting, 2004. IEDM Technical Digest. IEEE International*, pages 587–590, 2004. doi:[10.1109/IEDM.2004.1419228](https://doi.org/10.1109/IEDM.2004.1419228).
- I. Baek, D. Kim, M. Lee, H.-J. Kim, E. Yim, M. Lee, J. Lee, S. Ahn, S. Seo, J. Lee, J. Park, Y. Cha, S. Park, H. Kim, I. Yoo, U.-I. Chung, J. Moon, and B. Ryu. Multi-layer cross-point binary oxide resistive memory (OxRRAM) for post-NAND storage application. In *Electron Devices Meeting, 2005. IEDM Technical Digest. IEEE International*, pages 750–753, 2005. doi:[10.1109/IEDM.2005.1609462](https://doi.org/10.1109/IEDM.2005.1609462).
- A. Beck, J. G. Bednorz, C. Gerber, C. Rossel, and D. Widmer. Reproducible switching effect in thin oxide films for memory applications. *Applied Physics Letters*, 77(1): 139–141, 2000. doi:[10.1063/1.126902](https://doi.org/10.1063/1.126902).
- F. Bedeschi, R. Fackenthal, C. Resta, E. Donze, M. Jagasivamani, E. Buda, F. Pelizzier, D. Chow, A. Cabrini, G. Calvi, R. Faravelli, A. Fantini, G. Torelli, D. Mills, R. Gastaldi, and G. Casagrande. A bipolar-selected phase change memory featuring multi-level cell storage. *Solid-State Circuits, IEEE Journal of*, 44(1):217–227, 2009. doi:[10.1109/JSSC.2008.2006439](https://doi.org/10.1109/JSSC.2008.2006439).
- Y. Bengio, R. De Mori, G. Flammia, and R. Kompe. Global optimization of a neural network-hidden markov model hybrid. *Neural Networks, IEEE Transactions on*, 3(2):252–259, 1992. doi:[10.1109/72.125866](https://doi.org/10.1109/72.125866).
- G.-Q. Bi and M.-M. Poo. Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of Neuroscience*, 18:10464–10472, 1998.
- S. M. Bohte and M. C. Mozer. Reducing the variability of neural responses: A computational theory of spike-timing-dependent plasticity. *Neural Computation*, 19(2): 371–403, 2007. doi:[10.1162/neco.2007.19.2.371](https://doi.org/10.1162/neco.2007.19.2.371).
- J. Borghetti, Z. Li, J. Straznicky, X. Li, D. A. A. Ohlberg, W. Wu, D. R. Stewart, and R. S. Williams. A hybrid nanomemristor/transistor logic circuit capable of self-programming. *Proceedings of the National Academy of Sciences*, 106(6):1699–1703, 2009. doi:[10.1073/pnas.0806642106](https://doi.org/10.1073/pnas.0806642106).
- J. Borghetti, G. S. Snider, P. J. Kuekes, J. J. Yang, D. R. Stewart, and R. S. Williams. ‘Memristive’ switches enable ‘stateful’ logic operations via material implication. *Nature*, 464(7290):873–876, 2010. doi:[10.1038/nature08940](https://doi.org/10.1038/nature08940).
- J. S. Bowers. On the biological plausibility of grandmother cells: Implications for neural network theories in psychology and neuroscience. *Psychological Review*, 116(1):220–251, 2009. doi:[10.1037/a0014462](https://doi.org/10.1037/a0014462).
- M. Breitwisch, T. Nirschl, C. Chen, Y. Zhu, M. Lee, M. Lamorey, G. Burr, E. Joseph, A. Schrott, J. Philipp, R. Cheek, T. Happ, S. Chen, S. Zaidr, P. Flaitz, J. Bruley, R. Dasaka, B. Rajendran, S. Rossnage, M. Yang, Y. Chen, R. Bergmann, H. Lung, and C. Lam. Novel lithography-independent pore phase change memory. In *VLSI Technology, 2007 IEEE Symposium on*, pages 100–101, 2007. doi:[10.1109/VLSIT.2007.4339743](https://doi.org/10.1109/VLSIT.2007.4339743).

- M. J. Breitwisch, R. W. Cheek, C. H. Lam, D. Modha, and B. Rajendran. System for electronic learning synapse with spike-timing dependent plasticity using phase change memory. Patent Application, 11 2010. US 2010/0299297 A1.
- F. Brglez, C. Gloster, and G. Kedem. Hardware-based weighted random pattern generation for boundary scan. In *Test Conference, 1989. Proceedings. Meeting the Tests of Time., International*, pages 264–274, 1989. doi:[10.1109/TEST.1989.82307](https://doi.org/10.1109/TEST.1989.82307).
- Virtuoso Advanced Analysis Tools User Guide, Cadence 5.1.41 Spectre guide documents*. Cadence, 2008a. <http://www.cadence.com/>.
- Spectre/Matlab Tools User Guide, Cadence MMSIM 6.1 guide documents*. Cadence, 2008b. <http://www.cadence.com/>.
- X. Cao, X. Li, X. Gao, Y. Zhang, X. Liu, Q. Wang, and L. Chen. Effects of the compliance current on the resistive switching behavior of tio2 thin films. *Applied Physics A: Materials Science & Processing*, 97:883–887, 2009. doi:[10.1007/s00339-009-5351-7](https://doi.org/10.1007/s00339-009-5351-7).
- N. Carnevale and M. Hines. *The NEURON Book*. Cambridge University Press, New York, NY, USA, 2006. ISBN 0521843219.
- A. Chanthbouala, V. Garcia, R. O. Cherifi, K. Bouzehouane, S. Fusil, X. Moya, S. Xavier, H. Yamada, C. Deranlot, N. D. Mathur, M. Bibes, A. Barthélémy, and J. Grollier. A ferroelectric memristor. *Nature Materials*, 2012. doi:[10.1038/nmat3415](https://doi.org/10.1038/nmat3415).
- W. Chen, C. Lee, D. Chao, Y. Chen, F. Chen, C. Chen, R. Yen, M. Chen, W. Wang, T. Hsiao, J. Yeh, S. Chiou, M. Liu, T. Wang, L. Chein, C. Huang, N. Shih, L. Tu, D. Huang, T. Yu, M. Kao, and M.-J. Tsai. A novel cross-spacer phase change memory with ultra-small lithography independent contact area. In *Electron Devices Meeting, 2007. IEDM 2007. IEEE International*, pages 319–322, 2007. doi:[10.1109/IEDM.2007.4418935](https://doi.org/10.1109/IEDM.2007.4418935).
- Choudhary, S. Sloan, S. Fok, A. Neckar, E. Trautmann, P. Gao, T. Stewart, C. Elia-smith, and K. Boahen. Silicon neurons that compute. In *International Conference on Artificial Neural Networks*, 2012. In press.
- L. Chua. Memristor-the missing circuit element. *Circuit Theory, IEEE Transactions on*, 18(5):507–519, 1971. doi:[10.1109/TCT.1971.1083337](https://doi.org/10.1109/TCT.1971.1083337).
- L. Chua. Resistance switching memories are memristors. *Applied Physics A: Materials Science & Processing*, 102:765–783, 2011. doi:[10.1007/s00339-011-6264-9](https://doi.org/10.1007/s00339-011-6264-9).
- L. Chua and S. M. Kang. Memristive devices and systems. *Proceedings of the IEEE*, 64(2):209–223, 1976. doi:[10.1109/PROC.1976.10092](https://doi.org/10.1109/PROC.1976.10092).
- H. Chung, B. H. Jeong, B. Min, Y. Choi, B.-H. Cho, J. Shin, J. Kim, J. Sunwoo, J. min Park, Q. Wang, Y. jun Lee, S. Cha, D. Kwon, S. Kim, S. Kim, Y. Rho, M.-H. Park, J. Kim, I. Song, S. Jun, J. Lee, K. Kim, K. won Lim, W. ryul Chung, C. Choi, H. Cho, I. Shin, W. Jun, S. Hwang, K.-W. Song, K. Lee, S. whan Chang, W.-Y. Cho, J.-H. Yoo, and Y.-H. Jun. A 58nm 1.8V 1Gb PRAM with 6.4MB/s program BW. In *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International*, pages 500–502, 2011. doi:[10.1109/ISSCC.2011.5746415](https://doi.org/10.1109/ISSCC.2011.5746415).
- D. C. Cireşan, U. Meier, L. M. Gambardella, and S. Jürgen. Deep, big, simple neural nets for handwritten digit recognition. *Neural Computation*, 22(12):3207–3220, 2010. doi:[10.1162/NECO_a_00052](https://doi.org/10.1162/NECO_a_00052).

- Y. Dan and M. Ming Poo. Spike timing-dependent plasticity of neural circuits. *Neuron*, 44(1):23–30, 2004. doi:[10.1016/j.neuron.2004.09.007](https://doi.org/10.1016/j.neuron.2004.09.007).
- J. G. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *J. Opt. Soc. Am. A*, 2(7):1160–1169, 1985. doi:[10.1364/JOSAA.2.001160](https://doi.org/10.1364/JOSAA.2.001160).
- T. Delbruck. Frame-free dynamic digital vision. In *Intl. Symp. on Secure-Life Electronics, Advanced Electronics for Quality Life and Society*, pages 21–26, 2008. doi:[10.5167/uzh-17620](https://doi.org/10.5167/uzh-17620).
- A. Delorme, J. Gautrais, R. van Rullen, and S. Thorpe. SpikeNET: A simulator for modeling large networks of integrate and fire neurons. *Neurocomputing*, 26–27(0): 989–996, 1999. doi:[10.1016/S0925-2312\(99\)00095-8](https://doi.org/10.1016/S0925-2312(99)00095-8).
- A. Delorme, G. Richard, and M. Fabre-Thorpe. Ultra-rapid categorisation of natural scenes does not rely on colour cues: a study in monkeys and humans. *Vision Research*, 40(16):2187–200, 2000.
- A. Delorme, L. Perrinet, and S. Thorpe. Networks of integrate-and-fire neurons using rank order coding B: Spike timing dependent plasticity and emergence of orientation selectivity. *Neurocomputing*, 38:539–545, 2001.
- J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR09*, 2009.
- Y. H. Do, J. S. Kwak, Y. C. Bae, K. Jung, H. Im, and J. P. Hong. Hysteretic bipolar resistive switching characteristics in tio2/tio2-x multilayer homojunctions. *Applied Physics Letters*, 95(9):093507–093507–3, 2009. doi:[10.1063/1.3224179](https://doi.org/10.1063/1.3224179).
- DVS128 Dynamic Vision Sensor Silicon Retina data, Jan. 2011. <http://sourceforge.net/apps/trac/jaer/wiki/AER%20data>.
- A. Fantini, V. Sousa, L. Perniola, E. Gourvest, J. Bastien, S. Maitrejean, S. Braga, N. Pashkov, A. Bastard, B. Hyot, A. Roule, A. Persico, H. Feldis, C. Jahan, J. Nodin, D. Blachier, A. Toffoli, G. Reimbold, F. Fillot, F. Pierre, R. Annunziata, D. Ben-shael, P. Mazoyer, C. Vallée, T. Billon, J. Hazart, B. De Salvo, and F. Boulanger. N-doped GeTe as performance booster for embedded phase-change memories. In *Electron Devices Meeting (IEDM), 2010 IEEE International*, pages 29.1.1–29.1.4, 2010. doi:[10.1109/IEDM.2010.5703441](https://doi.org/10.1109/IEDM.2010.5703441).
- N. Frolet, M. Charbonneau, R. Tiron, J. Buckley, D. Mariolle, D. Boutry, R. Coppard, and B. De Salvo. Development of organic resistive memory for flexible electronics. In *MRS Proceedings*, volume 1430, pages mrss12–1430–e05–06, 2012. doi:[10.1557/opl.2012.1121](https://doi.org/10.1557/opl.2012.1121).
- S. B. Furber, S. Temple, and A. D. Brown. High-performance computing for systems of spiking neurons. *University of Pennsylvania Law Review*, 154(3):477+, 2006.
- C. Gasnier. Implémentation d’une transformée de fourier sur une architecture neuro-morphique. Rapport de stage de fin d’études, 2011.
- M.-O. Gewaltig and M. Diesmann. NEST (NEural Simulation Tool). *Scholarpedia*, 2(4):1430, 2007.
- B. R. Glasberg and B. C. Moore. Derivation of auditory filter shapes from notched-noise data. *Hearing Research*, 47(1–2):103–138, 1990. doi:[10.1016/0378-5955\(90\)90170-T](https://doi.org/10.1016/0378-5955(90)90170-T).

- D. Goodman and R. Brette. Brian: a simulator for spiking neural networks in python. *Frontiers in Neuroinformatics*, 2(5), 2008. doi:[10.3389/neuro.11.005.2008](https://doi.org/10.3389/neuro.11.005.2008).
- A. Gupta and L. Long. Character recognition using spiking neural networks. In *Neural Networks, 2007. IJCNN 2007. International Joint Conference on*, pages 53–58, 2007. doi:[10.1109/IJCNN.2007.4370930](https://doi.org/10.1109/IJCNN.2007.4370930).
- R. Guyonneau, R. VanRullen, and S. J. Thorpe. Temporal codes and sparse representations: A key to understanding rapid processing in the visual system. *Journal of Physiology-Paris*, 98(4–6):487–497, 2004. doi:[10.1016/j.jphysparis.2005.09.004](https://doi.org/10.1016/j.jphysparis.2005.09.004).
- R. Guyonneau, R. Vanrullen, and S. J. Thorpe. Neurons tune to the earliest spikes through stdp. *Neural Computation*, 17(4):859–879, 2005. doi:[10.1162/0899766053429390](https://doi.org/10.1162/0899766053429390).
- Y. Ha, J. Yi, H. Horii, J. Park, S. Joo, S. Park, U.-I. Chung, and J. Moon. An edge contact type cell for phase change RAM featuring very low power consumption. In *VLSI Technology, 2003. Digest of Technical Papers. 2003 Symposium on*, pages 175–176, 2003. doi:[10.1109/VLSIT.2003.1221142](https://doi.org/10.1109/VLSIT.2003.1221142).
- M. Hassoun. *Associative Neural Memories: Theory and Implementation*. Oxford Press, 1993.
- T. W. Hickmott. Low-frequency negative resistance in thin anodic oxide films. *Journal of Applied Physics*, 33(9):2669–2682, 1962. doi:[10.1063/1.1702530](https://doi.org/10.1063/1.1702530).
- K. Hynna and K. Boahen. Neuronal ion-channel dynamics in silicon. In *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*, page 4 pp., 2006. doi:[10.1109/ISCAS.2006.1693409](https://doi.org/10.1109/ISCAS.2006.1693409).
- IEEE. *IEEE Standard SystemC Language Reference Manual*, 2006. IEEE Std. 1666-2005.
- IEEE. *IEEE Standard VHDL Language Reference Manual*, 2008. IEEE Std. 1076-2008.
- D. Im, J. Lee, S. Cho, H. An, D. Kim, I. Kim, H. Park, D. Ahn, H. Horii, S. Park, U.-I. Chung, and J. Moon. A unified 7.5nm dash-type confined cell for high performance PRAM device. In *Electron Devices Meeting, 2008. IEDM 2008. IEEE International*, pages 1–4, 2008. doi:[10.1109/IEDM.2008.4796654](https://doi.org/10.1109/IEDM.2008.4796654).
- G. Indiveri, B. Linares-Barranco, T. J. Hamilton, A. van Schaik, R. Etienne-Cummings, T. Delbruck, S.-C. Liu, P. Dudek, P. Häfliger, S. Renaud, J. Schemmel, G. Cauwenberghs, J. Arthur, K. Hynna, F. Folowosele, S. Saighi, T. Serrano-Gotarredona, J. Wijekoon, Y. Wang, and K. Boahen. Neuromorphic silicon neuron circuits. *Frontiers in Neuroscience*, 5(00073), 2011. doi:[10.3389/fnins.2011.00073](https://doi.org/10.3389/fnins.2011.00073).
- H. Ishii, T. Shibata, H. Kosaka, and T. Ohmi. Hardware-backpropagation learning of neuron mos neural networks. In *Electron Devices Meeting, 1992. IEDM '92. Technical Digest., International*, pages 435–438, 1992. doi:[10.1109/IEDM.1992.307395](https://doi.org/10.1109/IEDM.1992.307395).
- ISO. *ISO/IEC 14882:2011 Information technology — Programming languages — C++*, 2011.
- ITRS. ITRS 2011 edition. Technical report, The International Technology Roadmap for Semiconductors, 2011. URL <http://www.itrs.net>.

- E. M. Izhikevich and N. S. Desai. Relating STDP to BCM. *Neural Computation*, 15(7):1511–1523, 2003. doi:[10.1162/089976603321891783](https://doi.org/10.1162/089976603321891783).
- B. L. Jackson, D. S. Modha, and B. Rajendran. Producing spike-timing dependent plasticity in an ultra-dense synapse cross-bar array. Patent Application, 06 2011. US 2011/0153533 A1.
- JAER Open Source Project, Jan. 2011. <http://jaer.wiki.sourceforge.net>.
- D. S. Jeong, H. Schroeder, and R. Waser. Coexistence of bipolar and unipolar resistive switching behaviors in a pt/tio₂/pt stack. *Electrochemical and Solid-State Letters*, 10(8):G51–G53, 2007. doi:[10.1149/1.2742989](https://doi.org/10.1149/1.2742989).
- X. Jin, A. Rast, F. Galluppi, S. Davies, and S. Furber. Implementing spike-timing-dependent plasticity on spinnaker neuromorphic hardware. In *Neural Networks (IJCNN), The 2010 International Joint Conference on*, pages 1–8, 2010. doi:[10.1109/IJCNN.2010.5596372](https://doi.org/10.1109/IJCNN.2010.5596372).
- S. H. Jo and W. Lu. Cmos compatible nanoscale nonvolatile resistance switching memory. *Nano Letters*, 8(2):392–397, 2008. doi:[10.1021/nl073225h](https://doi.org/10.1021/nl073225h).
- S. H. Jo, K.-H. Kim, and W. Lu. High-density crossbar arrays based on a si memristive system. *Nano Letters*, 9(2):870–874, 2009a. doi:[10.1021/nl8037689](https://doi.org/10.1021/nl8037689).
- S. H. Jo, K.-H. Kim, and W. Lu. Programmable resistance switching in nanoscale two-terminal devices. *Nano Letters*, 9(1):496–500, 2009b. doi:[10.1021/nl803669s](https://doi.org/10.1021/nl803669s).
- S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu. Nanoscale memristor device as synapse in neuromorphic systems. *Nano Letters*, 10(4):1297–1301, 2010. doi:[10.1021/nl904092h](https://doi.org/10.1021/nl904092h).
- A. Joubert, B. Belhadj, and R. Heliot. A robust and compact 65 nm LIF analog neuron for computational purposes. In *New Circuits and Systems Conference (NEWCAS), 2011 IEEE 9th International*, pages 9–12, 2011. doi:[10.1109/NEWCAS.2011.5981206](https://doi.org/10.1109/NEWCAS.2011.5981206).
- A. Joubert, B. Belhadj, O. Temam, and R. Heliot. Hardware spiking neurons design: Analog or digital? In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pages 1–5, 2012. doi:[10.1109/IJCNN.2012.6252600](https://doi.org/10.1109/IJCNN.2012.6252600).
- E. R. Kandel, J. H. Schwartz, and T. M. Jessell. *Principles of Neural Science*. McGraw-Hill Medical, 4th edition, 2000. ISBN 0071120009.
- I. V. Karpov, M. Mitra, D. Kau, G. Spadini, Y. A. Kryukov, and V. G. Karpov. Fundamental drift of parameters in chalcogenide phase change memory. *Journal of Applied Physics*, 102(12):124503–124503–6, 2007. doi:[10.1063/1.2825650](https://doi.org/10.1063/1.2825650).
- I. Kim, S. Cho, D. Im, E. Cho, D. Kim, G. Oh, D. Ahn, S. Park, S. Nam, J. Moon, and C. Chung. High performance PRAM cell scalable to sub-20nm technology with below 4F² cell size, extendable to DRAM applications. In *VLSI Technology (VLSIT), 2010 Symposium on*, pages 203–204, 2010. doi:[10.1109/VLSIT.2010.5556228](https://doi.org/10.1109/VLSIT.2010.5556228).
- K.-H. Kim, S. Gaba, D. Wheeler, J. M. Cruz-Albrecht, T. Hussain, N. Srinivasa, and W. Lu. A functional hybrid memristor crossbar-array/cmos system for data storage and neuromorphic applications. *Nano Letters*, 12(1):389–395, 2012. doi:[10.1021/nl203687n](https://doi.org/10.1021/nl203687n).

- H. Kosaka, T. Shibata, H. Ishii, and T. Ohmi. An excellent weight-updating-linearity EEPROM synapse memory cell for self-learning neuron-MOS neural networks. *Electron Devices, IEEE Transactions on*, 42(1):135–143, 1995. doi:[10.1109/16.370025](https://doi.org/10.1109/16.370025).
- M. Kozicki, M. Park, and M. Mitkova. Nanoscale memory elements based on solid-state electrolytes. *Nanotechnology, IEEE Transactions on*, 4(3):331–338, 2005. doi:[10.1109/TNANO.2005.846936](https://doi.org/10.1109/TNANO.2005.846936).
- M. N. Kozicki and W. C. West. Programmable metallization cell structure and method of making same. Patent Application, 06 1998. US 5761115.
- P. J. Kuekes, W. Robinett, G. Seroussi, and R. S. Williams. Defect-tolerant interconnect to nanoelectronic circuits: internally redundant demultiplexers based on error-correcting codes. *Nanotechnology*, 16(6):869, 2005.
- M. Kund, G. Beitel, C.-U. Pinnow, T. Rohr, J. Schumann, R. Symanczyk, K.-D. Ufert, and G. Muller. Conductive bridging RAM (CBRAM): an emerging non-volatile memory technology scalable to sub 20nm. In *Electron Devices Meeting, 2005. IEDM Technical Digest. IEEE International*, pages 754–757, 2005. doi:[10.1109/IEDM.2005.1609463](https://doi.org/10.1109/IEDM.2005.1609463).
- D. Kuzum, R. G. D. Jeyasingh, B. Lee, and H.-S. P. Wong. Nanoelectronic programmable synapses based on phase change materials for brain-inspired computing. *Nano Letters*, 12(5):2179–2186, 2012. doi:[10.1021/nl201040y](https://doi.org/10.1021/nl201040y).
- Q. Lai, Z. Li, L. Zhang, X. Li, W. F. Stickle, Z. Zhu, Z. Gu, T. I. Kamins, R. S. Williams, and Y. Chen. An organic/si nanowire hybrid field configurable transistor. *Nano Letters*, 8(3):876–880, 2008. doi:[10.1021/nl073112y](https://doi.org/10.1021/nl073112y).
- Q. V. Le, R. Monga, M. Devin, G. Corrado, K. Chen, M. Ranzato, J. Dean, and A. Y. Ng. Building high-level features using large scale unsupervised learning. *CoRR*, abs/1112.6209, 2011.
- Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi:[10.1109/5.726791](https://doi.org/10.1109/5.726791).
- J. Lee, H. Park, S. Cho, Y. Park, B. Bae, J. Park, J. Park, H. An, J. Bae, D. Ahn, Y. Kim, H. Horii, S. Song, J. Shin, S. Park, H. Kim, U.-I. Chung, J. Moon, and B. Ryu. Highly scalable phase change memory with CVD $G_eS_bT_e$ for sub 50nm generation. In *VLSI Technology, 2007 IEEE Symposium on*, pages 102–103, 2007a. doi:[10.1109/VLSIT.2007.4339744](https://doi.org/10.1109/VLSIT.2007.4339744).
- J. H. Lee and K. K. Likharev. Defect-tolerant nanoelectronic pattern classifiers. *International Journal of Circuit Theory and Applications*, 35(3):239–264, 2007. doi:[10.1002/cta.410](https://doi.org/10.1002/cta.410).
- K.-J. Lee, B.-H. Cho, W.-Y. Cho, S. Kang, B.-G. Choi, H.-R. Oh, C.-S. Lee, H.-J. Kim, J.-M. Park, Q. Wang, M.-H. Park, Y.-H. Ro, J.-Y. Choi, K.-S. Kim, Y.-R. Kim, I.-C. Shin, K.-W. Lim, H.-K. Cho, C.-H. Choi, W.-R. Chung, D.-E. Kim, K.-S. Yu, G.-T. Jeong, H.-S. Jeong, C.-K. Kwak, C.-H. Kim, and K. Kim. A 90nm 1.8v 512Mb diode-switch PRAM with 266MB/s read throughput. In *Solid-State Circuits Conference, 2007. ISSCC 2007. Digest of Technical Papers. IEEE International*, pages 472–616, 2007b. doi:[10.1109/ISSCC.2007.373499](https://doi.org/10.1109/ISSCC.2007.373499).

- G. Leuba and R. Kraftsik. Changes in volume, surface estimate, three-dimensional shape and total number of neurons of the human primary visual cortex from midgestation until old age. *Anatomy and Embryology*, 190(4):351–366, 1994. doi:[10.1007/BF00187293](https://doi.org/10.1007/BF00187293).
- Y. Li, A. Sinitskii, and J. M. Tour. Electronic two-terminal bistable graphitic memories. *Nature Materials*, 7(12):966–971, 2008. doi:[10.1038/nmat2331](https://doi.org/10.1038/nmat2331).
- J. Liang and H.-S. Wong. Cross-point memory array without cell selectors - device characteristics and data storage pattern dependencies. *Electron Devices, IEEE Transactions on*, 57(10):2531–2538, 2010. doi:[10.1109/TED.2010.2062187](https://doi.org/10.1109/TED.2010.2062187).
- J. Liang, R. Jeyasingh, H.-Y. Chen, and H.-S. Wong. A 1.4 μ A reset current phase change memory cell with integrated carbon nanotube electrodes for cross-point memory application. In *VLSI Technology (VLSIT), 2011 Symposium on*, pages 100–101, 2011.
- P. Lichtsteiner, C. Posch, and T. Delbruck. A 128×128 120 dB 15 μ s latency asynchronous temporal contrast vision sensor. *Solid-State Circuits, IEEE Journal of*, 43(2):566–576, 2008. doi:[10.1109/JSSC.2007.914337](https://doi.org/10.1109/JSSC.2007.914337).
- C.-Y. Lin, C.-Y. Liu, C.-C. Lin, and T. Tseng. Current status of resistive non-volatile memories. *Journal of Electroceramics*, 21:61–66, 2008. ISSN 1385-3449. doi:[10.1007/s10832-007-9081-y](https://doi.org/10.1007/s10832-007-9081-y).
- B. Linares-Barranco and T. Serrano-Gotarredona. Memristance can explain spike-time dependent plasticity in neural synapses. *Nature Precedings*, 2009.
- E. Linn, R. Rosezin, C. Kugeler, and R. Waser. Complementary resistive switches for passive nanocrossbar memories. *Nature Materials*, 9(5):403–406, 2010. doi:[10.1038/nmat2748](https://doi.org/10.1038/nmat2748).
- J. Lisman and N. Spruston. Questions about STDP as a general model of synaptic plasticity. *Frontiers in Synaptic Neuroscience*, 2(140), 2010. doi:[10.3389/fnsyn.2010.00140](https://doi.org/10.3389/fnsyn.2010.00140).
- S.-C. Liu, A. van Schaik, B. Minch, and T. Delbruck. Event-based 64-channel binocular silicon cochlea with Q enhancement mechanisms. In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pages 2027–2030, 2010. doi:[10.1109/ISCAS.2010.5537164](https://doi.org/10.1109/ISCAS.2010.5537164).
- P. Livi and G. Indiveri. A current-mode conductance-based silicon neuron for address-event neuromorphic systems. In *Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on*, pages 2898–2901, 2009. doi:[10.1109/ISCAS.2009.5118408](https://doi.org/10.1109/ISCAS.2009.5118408).
- W. Maass. Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 10(9):1659–1671, 1997. doi:[10.1016/S0893-6080\(97\)00011-7](https://doi.org/10.1016/S0893-6080(97)00011-7).
- W. Maass, T. Natschläger, and H. Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14(11):2531–2560, 2002. doi:[10.1162/089976602760407955](https://doi.org/10.1162/089976602760407955).
- N. Macmillan and C. Creelman. *Detection Theory: A User's Guide*. Taylor & Francis, 2004. ISBN 9780805842302.

- H. Markram, J. Lübke, M. Frotscher, and B. Sakmann. Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science*, 275(5297):213–215, 1997. doi:[10.1126/science.275.5297.213](https://doi.org/10.1126/science.275.5297.213).
- T. Masquelier and S. J. Thorpe. Unsupervised learning of visual features through spike timing dependent plasticity. *PLoS Comput Biol*, 3(2):e31, 2007. doi:[10.1371/journal.pcbi.0030031](https://doi.org/10.1371/journal.pcbi.0030031).
- T. Masquelier, R. Guyonneau, and S. J. Thorpe. Spike timing dependent plasticity finds the start of repeating patterns in continuous spike trains. *PLoS ONE*, 3(1):e1377, 2008. doi:[10.1371/journal.pone.0001377](https://doi.org/10.1371/journal.pone.0001377).
- T. Masquelier, R. Guyonneau, and S. J. Thorpe. Competitive stdp-based spike pattern learning. *Neural Comput.*, 21:1259–1276, 2009. doi:[10.1162/neco.2008.06-08-804](https://doi.org/10.1162/neco.2008.06-08-804).
- R. Meixner, H. Gobel, H. Qiu, C. Ucurum, W. Klix, R. Stenzel, F. Yildirim, W. Bauhofer, and W. Krautschneider. A physical-based PSPICE compact model for poly(3-hexylthiophene) organic field-effect transistors. *Electron Devices, IEEE Transactions on*, 55(7):1776–1781, 2008. doi:[10.1109/TED.2008.925339](https://doi.org/10.1109/TED.2008.925339).
- D. Modha and R. S. Shenoy. Electronic learning synapse with spike-timing dependent plasticity using unipolar memory-switching elements. Patent Application, 11 2010. US 2010/0299296 A1.
- B. Muthuswamy and P. Kokate. Memristor-Based Chaotic Circuits. *IETE Technical Review*, 26(6):417–429, 2009. doi:[10.4103/0256-4602.57827](https://doi.org/10.4103/0256-4602.57827).
- B. Nessler, M. Pfeiffer, and W. Maass. STDP enables spiking neurons to detect hidden causes of their inputs. In *Advances in Neural Information Processing Systems*, volume 22, pages 1357–1365, 2010.
- T. Nirschl, J. Phipp, T. Happ, G. Burr, B. Rajendran, M.-H. Lee, A. Schrott, M. Yang, M. Breitwisch, C.-F. Chen, E. Joseph, M. Lamorey, R. Cheek, S.-H. Chen, S. Zaidi, S. Raoux, Y. Chen, Y. Zhu, R. Bergmann, H.-L. Lung, and C. Lam. Write strategies for 2 and 4-bit multi-level phase-change memory. In *Electron Devices Meeting, 2007. IEDM 2007. IEEE International*, pages 461–464, 2007. doi:[10.1109/IEDM.2007.4418973](https://doi.org/10.1109/IEDM.2007.4418973).
- M. Nishiyama, K. Hong, K. Mikoshiba, M.-m. Poo, and K. Kato. Calcium stores regulate the polarity and input specificity of synaptic modification. *Nature*, 408:584–588, 2000. doi:[10.1038/35046067](https://doi.org/10.1038/35046067).
- D. Norman. *The Brain That Changes Itself*. United States: Viking Press, 2007.
- C. Novembre, D. Guérin, K. Lmimouni, C. Gamrat, and D. Vuillaume. Gold nanoparticle-pentacene memory transistors. *Applied Physics Letters*, 92(10):103314, 2008. doi:[10.1063/1.2896602](https://doi.org/10.1063/1.2896602).
- B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996. doi:[10.1038/381607a0](https://doi.org/10.1038/381607a0).
- G. Palma, E. Vianello, C. Cagli, G. Molas, M. Reyboz, P. Blaise, B. De Salvo, F. Longnos, and F. Dahmani. Experimental investigation and empirical modeling of the set and reset kinetics of Ag-GeS₂ conductive bridging memories. In *Memory Workshop (IMW), 2012 4th IEEE International*, pages 1–4, 2012. doi:[10.1109/IMW.2012.6213680](https://doi.org/10.1109/IMW.2012.6213680).

- N. Papandreou, H. Pozidis, T. Mittelholzer, G. Close, M. Breitwisch, C. Lam, and E. Eleftheriou. Drift-tolerant multilevel phase-change memory. In *Memory Workshop (IMW), 2011 3rd IEEE International*, pages 1–4, 2011. doi:[10.1109/IMW.2011.5873231](https://doi.org/10.1109/IMW.2011.5873231).
- I. Pavlov. Conditioned reflexes: An investigation of the physiological activity of the cerebral cortex. translated and edited by g. v. anrep. London: Oxford University Press, 1927.
- F. Pellizzer, A. Benvenuti, B. Gleixner, Y. Kim, B. Johnson, M. Magistretti, T. Marangon, A. Pirovano, R. Bez, and G. Atwood. A 90nm phase change memory technology for stand-alone non-volatile memory applications. In *VLSI Technology, 2006. Digest of Technical Papers. 2006 Symposium on*, pages 122–123, 2006. doi:[10.1109/VLSIT.2006.1705247](https://doi.org/10.1109/VLSIT.2006.1705247).
- C. Peng, L. Cheng, and M. Mansuripur. Experimental and theoretical investigations of laser-induced crystallization and amorphization in phase-change optical recording media. *Journal of Applied Physics*, 82(9):4183–4191, 1997. doi:[10.1063/1.366220](https://doi.org/10.1063/1.366220).
- Y. Pershin and M. Di Ventra. Experimental demonstration of associative memory with memristive neural networks. *Neural Networks*, 23(7):881–886, 2010a. doi:[10.1016/j.neunet.2010.05.001](https://doi.org/10.1016/j.neunet.2010.05.001).
- Y. Pershin and M. Di Ventra. Practical approach to programmable analog circuits with memristors. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 57(8):1857–1864, 2010b. doi:[10.1109/TCSI.2009.2038539](https://doi.org/10.1109/TCSI.2009.2038539).
- Y. V. Pershin, S. La Fontaine, and M. Di Ventra. Memristive model of amoeba learning. *Phys. Rev. E*, 80:021926, 2009. doi:[10.1103/PhysRevE.80.021926](https://doi.org/10.1103/PhysRevE.80.021926).
- M. D. Pickett, D. B. Strukov, J. L. Borghetti, J. J. Yang, G. S. Snider, D. R. Stewart, and R. S. Williams. Switching dynamics in titanium dioxide memristive devices. *Journal of Applied Physics*, 106(7):074508, 2009. doi:[10.1063/1.3236506](https://doi.org/10.1063/1.3236506).
- A. Pirovano, F. Pellizzer, I. Tortorelli, R. Harrigan, M. Magistretti, P. Petruzza, E. Varesi, D. Erbetta, T. Marangon, F. Bedeschi, R. Fackenthal, G. Atwood, and R. Bez. Self-aligned μ trench phase-change memory cell architecture for 90nm technology and beyond. In *Solid State Device Research Conference, 2007. ESSDERC 2007. 37th European*, pages 222–225, 2007. doi:[10.1109/ESSDERC.2007.4430918](https://doi.org/10.1109/ESSDERC.2007.4430918).
- G. Potamianos, C. Neti, G. Gravier, A. Garg, and A. Senior. Recent advances in the automatic recognition of audiovisual speech. *Proceedings of the IEEE*, 91(9):1306–1326, 2003. doi:[10.1109/JPROC.2003.817150](https://doi.org/10.1109/JPROC.2003.817150).
- D. Povey, L. Burget, M. Agarwal, P. Akyazi, F. Kai, A. Ghoshal, O. Glembek, N. Goel, M. Karafiát, A. Rastrow, R. C. Rose, P. Schwarz, and S. Thomas. The subspace gaussian mixture model—a structured model for speech recognition. *Computer Speech & Language*, 25(2):404–439, 2011. doi:[10.1016/j.csl.2010.06.003](https://doi.org/10.1016/j.csl.2010.06.003).
- J. Pérez-Carrasco, C. Serrano, B. Acha, T. Serrano-Gotarredona, and B. Linares-Barranco. Spike-based convolutional network for real-time processing. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 3085–3088, 2010. doi:[10.1109/ICPR.2010.756](https://doi.org/10.1109/ICPR.2010.756).
- D. Purves, M. Jeannerod, and J. Coquery. *Neurosciences*. Neurosciences & cognition. De Boeck Supérieur, 2005. ISBN 9782804147976.

- R. Quian Quiroga, L. Reddy, G. Kreiman, C. Koch, and I. Fried. Invariant visual representation by single neurons in the human brain. *Nature*, 435:1102–1107, 2005. doi:[10.1038/nature03687](https://doi.org/10.1038/nature03687).
- C. Ramana, M. Moodely, V. Kannan, A. Maity, J. Jayaramudu, and W. Clarke. Fabrication of stable low voltage organic bistable memory device. *Sensors and Actuators B: Chemical*, 161(1):684 – 688, 2012. doi:[10.1016/j.snb.2011.11.012](https://doi.org/10.1016/j.snb.2011.11.012).
- W. Robinett, M. Pickett, J. Borghetti, Q. Xia, G. S. Snider, G. Medeiros-Ribeiro, and R. S. Williams. A memristor-based nonvolatile latch circuit. *Nanotechnology*, 21(23):235203, 2010.
- F. Rosenblatt. The perceptron: probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–408, 1958.
- A. Roy. Discovery of concept cells in the human brain – could it change our science? *Natural Intelligence: the INNS Magazine*, 1(1):23–29, 2011.
- J. Rubin, D. D. Lee, and H. Sompolinsky. Equilibrium Properties of Temporally Asymmetric Hebbian Plasticity. *Physical Review Letters*, 86(2):364+, 2001. doi:[10.1103/PhysRevLett.86.364](https://doi.org/10.1103/PhysRevLett.86.364).
- D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In D. E. Rumelhart and J. L. McClelland, editors, *Parallel distributed processing: explorations in the microstructure of cognition, vol. 1*, pages 318–362. MIT Press, Cambridge, MA, USA, 1986.
- Y. Sasago, M. Kinoshita, T. Morikawa, K. Kurotsuchi, S. Hanzawa, T. Mine, A. Shima, Y. Fujisaki, H. Kume, H. Moriya, N. Takaura, and K. Torii. Cross-point phase change memory with 4F2 cell size driven by low-contact-resistivity poly-Si diode. In *VLSI Technology, 2009 Symposium on*, pages 24–25, 2009.
- Y. Sato, K. Tsunoda, K. Kinoshita, H. Noshiro, M. Aoki, and Y. Sugiyama. Sub-100- μ A reset current of nickel oxide resistive memory through control of filamentary conductance by current limit of mosfet. *Electron Devices, IEEE Transactions on*, 55(5):1185–1191, 2008. doi:[10.1109/TED.2008.919385](https://doi.org/10.1109/TED.2008.919385).
- A. Sawa. Resistive switching in transition metal oxides. *Materials Today*, 11(6):28–36, 2008. doi:[10.1016/S1369-7021\(08\)70119-6](https://doi.org/10.1016/S1369-7021(08)70119-6).
- J. Schemmel, A. Grubl, K. Meier, and E. Mueller. Implementing synaptic plasticity in a vlsi spiking neural network model. In *Neural Networks, 2006. IJCNN '06. International Joint Conference on*, pages 1–6, 2006. doi:[10.1109/IJCNN.2006.246651](https://doi.org/10.1109/IJCNN.2006.246651).
- J. Schemmel, D. Brüderle, A. Gröbl, M. Hock, K. Meier, and S. Millner. A wafer-scale neuromorphic hardware system for large-scale neural modeling. In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pages 1947–1950, 2010. doi:[10.1109/ISCAS.2010.5536970](https://doi.org/10.1109/ISCAS.2010.5536970).
- C. Schindler, S. Therman, R. Waser, and M. Kozicki. Bipolar and unipolar resistive switching in cu-doped SiO₂. *Electron Devices, IEEE Transactions on*, 54(10):2762–2768, 2007. doi:[10.1109/TED.2007.904402](https://doi.org/10.1109/TED.2007.904402).
- C. Schindler, M. Weides, M. N. Kozicki, and R. Waser. Low current resistive switching in Cu-SiO₂ cells. *Applied Physics Letters*, 92(12):122910–122910–3, 2008. doi:[10.1063/1.2903707](https://doi.org/10.1063/1.2903707).

- P. Schrogmeier, M. Angerbauer, S. Dietrich, M. Ivanov, H. Honigschmid, C. Liaw, M. Markert, R. Symanczyk, L. Altimime, S. Bournat, and G. Muller. Time discrete voltage sensing and iterative programming control for a 4F2 multilevel CBRAM. In *VLSI Circuits, 2007 IEEE Symposium on*, pages 186–187, 2007. doi:[10.1109/VLSIC.2007.4342708](https://doi.org/10.1109/VLSIC.2007.4342708).
- G. Servalli. A 45nm generation phase change memory technology. In *Electron Devices Meeting (IEDM), 2009 IEEE International*, pages 1–4, 2009. doi:[10.1109/IEDM.2009.5424409](https://doi.org/10.1109/IEDM.2009.5424409).
- M. Sharad, C. Augustine, G. Panagopoulos, and K. Roy. Proposal for neuromorphic hardware using spin devices. *CoRR*, abs/1206.3227, 2012.
- M. Slaney. An efficient implementation of the patterson-holdsworth auditory filter bank, apple computer TR #35. Technical report, Perception Group-Advanced Technology Group, Apple Computer, 1993.
- G. Snider. Spike-timing-dependent learning in memristive nanodevices. In *Nanoscale Architectures, 2008. NANOARCH 2008. IEEE International Symposium on*, pages 85–92, 2008. doi:[10.1109/NANOARCH.2008.4585796](https://doi.org/10.1109/NANOARCH.2008.4585796).
- G. Snider, R. Amerson, D. Carter, H. Abdalla, M. Qureshi, J. Léveillé, M. Versace, H. Ames, S. Patrick, B. Chandler, A. Gorchetchnikov, and E. Mingolla. From synapses to circuitry: Using memristive memory to explore the electronic brain. *Computer*, 44(2):21–28, 2011. doi:[10.1109/MC.2011.48](https://doi.org/10.1109/MC.2011.48).
- G. S. Snider. Self-organized computation with unreliable, memristive nanodevices. *Nanotechnology*, 18(36):365202, 2007. doi:[10.1088/0957-4484/18/36/365202](https://doi.org/10.1088/0957-4484/18/36/365202).
- G. S. Snider and R. S. Williams. Nano/CMOS architectures using a field-programmable nanowire interconnect. *Nanotechnology*, 18(3):035204, 2007. doi:[10.1088/0957-4484/18/3/035204](https://doi.org/10.1088/0957-4484/18/3/035204).
- K. Sonoda, A. Sakai, M. Moniwa, K. Ishikawa, O. Tsuchiya, and Y. Inoue. A compact model of phase-change memory based on rate equations of crystallization and amorphization. *Electron Devices, IEEE Transactions on*, 55(7):1672–1681, 2008. doi:[10.1109/TED.2008.923740](https://doi.org/10.1109/TED.2008.923740).
- R. Stein. Some models of neuronal variability. *Biophysical Journal*, 7(1):37–68, 1967. doi:[10.1016/S0006-3495\(67\)86574-3](https://doi.org/10.1016/S0006-3495(67)86574-3).
- D. B. Strukov and K. K. Likharev. Cmol fpga: a reconfigurable architecture for hybrid digital circuits with two-terminal nanodevices. *Nanotechnology*, 16(6):888, 2005.
- D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams. The missing memristor found. *Nature*, 453(7191):80–83, 2008. doi:[10.1038/nature06932](https://doi.org/10.1038/nature06932).
- D. B. Strukov, J. L. Borghetti, and R. S. Williams. Coupled ionic and electronic transport model of thin-film semiconductor memristive behavior. *Small*, 5(9):1058–1063, 2009. doi:[10.1002/sml.200801323](https://doi.org/10.1002/sml.200801323).
- M. Suri, V. Sousa, L. Perniola, D. Vuillaume, and B. DeSalvo. Phase change memory for synaptic plasticity application in neuromorphic systems. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 619–624, 2011. doi:[10.1109/IJCNN.2011.6033278](https://doi.org/10.1109/IJCNN.2011.6033278).

- B. Swaroop, W. West, G. Martinez, M. Kozicki, and L. Akers. Programmable current mode hebbian learning neural network using programmable metallization cell. In *Circuits and Systems, 1998. ISCAS '98. Proceedings of the 1998 IEEE International Symposium on*, volume 3, pages 33–36, 1998. doi:[10.1109/ISCAS.1998.703888](https://doi.org/10.1109/ISCAS.1998.703888).
- K. Szot, W. Speier, G. Bihlmayer, and R. Waser. Switching the electrical resistance of individual dislocations in single-crystalline SrTiO₃. *Nature Materials*, 5(4):312–320, 2006. doi:[10.1038/nmat1614](https://doi.org/10.1038/nmat1614).
- S. Tam, M. Holler, J. Brauch, A. Pine, A. Peterson, S. Anderson, and S. Deiss. A reconfigurable multi-chip analog neural network: recognition and back-propagation training. In *Neural Networks, 1992. IJCNN., International Joint Conference on*, volume 2, pages 625–630, 1992. doi:[10.1109/IJCNN.1992.226918](https://doi.org/10.1109/IJCNN.1992.226918).
- S. Thorpe, D. Fize, and C. Marlot. Speed of processing in the human visual system. *Nature*, 381(6582):520–522, 1996. doi:[10.1038/381520a0](https://doi.org/10.1038/381520a0).
- S. Thorpe, A. Brilhault, and J.-A. Perez-Carrasco. Suggestions for a biologically inspired spiking retina using order-based coding. In *Circuits and Systems (IS-CAS), Proceedings of 2010 IEEE International Symposium on*, pages 265–268, 2010. doi:[10.1109/ISCAS.2010.5537898](https://doi.org/10.1109/ISCAS.2010.5537898).
- S. J. Thorpe. Spike arrival tiems: A highly efficient coding scheme for neural networks. In G. Eckmiller and G. Hauske, editors, *Parallel processing in neural systems*, pages 91–94. Elsevier, North-Holland, 1990.
- S. J. Thorpe and M. Imbert. Biological constraints on connectionist modelling. In *Connectionism in Perspective*, pages 63–92. Elsevier, 1989.
- S. J. Thorpe, R. Guyonneau, N. Guilbaud, J.-M. Allegraud, and R. VanRullen. Spikenet: real-time visual processing with one spike per neuron. *Neurocomputing*, 58–60(0):857–864, 2004. doi:[10.1016/j.neucom.2004.01.138](https://doi.org/10.1016/j.neucom.2004.01.138).
- A. Torralba. How many pixels make an image? *Visual Neuroscience*, 26:123–131, 2009. doi:[10.1017/S0952523808080930](https://doi.org/10.1017/S0952523808080930).
- M. Tsodyks, K. Pawelzik, and H. Markram. Neural networks with dynamic synapses. *Neural Computation*, 10:821–835, 1998. doi:[10.1162/089976698300017502](https://doi.org/10.1162/089976698300017502).
- K. Tsuchida, T. Inaba, K. Fujita, Y. Ueda, T. Shimizu, Y. Asao, T. Kajiyama, M. Iwayama, K. Sugiura, S. Ikegawa, T. Kishi, T. Kai, M. Amano, N. Shimomura, H. Yoda, and Y. Watanabe. A 64Mb MRAM with clamped-reference and adequate-reference schemes. In *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, pages 258–259, 2010. doi:[10.1109/ISSCC.2010.5433948](https://doi.org/10.1109/ISSCC.2010.5433948).
- H. Tuckwell. *Introduction to Theoretical Neurobiology: Volume 1, Linear Cable Theory and Dendritic Structure*. Cambridge Studies in Mathematical Biology. Cambridge University Press, 1988. ISBN 9780521022224.
- R. Van Rullen and S. J. Thorpe. Rate coding versus temporal order coding: What the retinal ganglion cells tell the visual cortex. *Neural Computation*, 13(6):1255–1283, 2001. doi:[10.1162/08997660152002852](https://doi.org/10.1162/08997660152002852).
- R. VanRullen and S. J. Thorpe. Surfing a spike wave down the ventral stream. *Vision Research*, 42(23):2593–2615, 2002. doi:[10.1016/S0042-6989\(02\)00298-5](https://doi.org/10.1016/S0042-6989(02)00298-5).

- M. Versace and B. Chandler. Moneta: A mind made from memristors. *IEEE Spectrum*, 47(12):30–37, 2010.
- P. Viola and M. J. Jones. Robust real-time face detection. *Int. J. Comput. Vision*, 57(2):137–154, 2004. doi:[10.1023/B:VISI.0000013087.49260.fb](https://doi.org/10.1023/B:VISI.0000013087.49260.fb).
- R. Waser and M. Aono. Nanoionics-based resistive switching memories. *Nature Materials*, 6(11):833–840, 2007. doi:[10.1038/nmat2023](https://doi.org/10.1038/nmat2023).
- H. Wersing and E. Körner. Learning optimized features for hierarchical models of invariant object recognition. *Neural Computation*, 15(7):1559–1588, 2003. doi:[10.1162/089976603321891800](https://doi.org/10.1162/089976603321891800).
- B. Widrow, W. H. Pierce, and J. Angell. Birth, life, and death in microelectronic systems. *Military Electronics, IRE Transactions on*, MIL-5(3):191–201, 1961. doi:[10.1109/IRET-MIL.1961.5008348](https://doi.org/10.1109/IRET-MIL.1961.5008348).
- S. Wimbauer, W. Gerstner, and J. van Hemmen. Emergence of spatiotemporal receptive fields and its application to motion detection. *Biological Cybernetics*, 72:81–92, 1994. doi:[10.1007/BF00206240](https://doi.org/10.1007/BF00206240).
- G. M. Wittenberg and S. S.-H. Wang. Malleability of spike-timing-dependent plasticity at the CA3-CA1 synapse. *The Journal of Neuroscience*, 26(24):6610–6617, 2006. doi:[10.1523/JNEUROSCI.5388-05.2006](https://doi.org/10.1523/JNEUROSCI.5388-05.2006).
- M. A. Woodin, K. Ganguly, and M. ming Poo. Coincident pre- and postsynaptic activity modifies gabaergic synapses by postsynaptic changes in cl^- transporter activity. *Neuron*, 39(5):807 – 820, 2003. doi:[10.1016/S0896-6273\(03\)00507-5](https://doi.org/10.1016/S0896-6273(03)00507-5).
- M. Wuttig and N. Yamada. Phase-change materials for rewriteable data storage. *Nature Materials*, 6:824–832, 2007. doi:[10.1038/nmat2009](https://doi.org/10.1038/nmat2009).
- Q. Xia, W. Robinett, M. W. Cumbie, N. Banerjee, T. J. Cardinali, J. J. Yang, W. Wu, X. Li, W. M. Tong, D. B. Strukov, G. S. Snider, G. Medeiros-Ribeiro, and R. S. Williams. Memristor-cmos hybrid integrated circuits for reconfigurable logic. *Nano Letters*, 9(10):3640–3645, 2009. doi:[10.1021/nl901874j](https://doi.org/10.1021/nl901874j).
- F. Xiong, A. D. Liao, D. Estrada, and E. Pop. Low-power switching of phase-change materials with carbon nanotube electrodes. *Science*, 332(6029):568–570, 2011. doi:[10.1126/science.1201938](https://doi.org/10.1126/science.1201938).
- J. J. Yang, M. D. Pickett, X. Li, D. A. A. Ohlberg, D. R. Stewart, and R. S. Williams. Memristive switching mechanism for metal/oxide/metal nanodevices. *Nature Nanotechnology*, 3(7):429–433, 2008. doi:[10.1038/nnano.2008.160](https://doi.org/10.1038/nnano.2008.160).
- J. J. Yang, M.-X. Zhang, M. D. Pickett, F. Miao, J. P. Strachan, W.-D. Li, W. Yi, D. A. A. Ohlberg, B. J. Choi, W. Wu, J. H. Nickel, G. Medeiros-Ribeiro, and R. S. Williams. Engineering nonlinearity into memristors for passive crossbar applications. *Applied Physics Letters*, 100(11):113501, 2012. doi:[10.1063/1.3693392](https://doi.org/10.1063/1.3693392).
- J. J. Yon, E. Mottin, L. Biancardini, L. Letellier, and J. L. Tissot. Infrared microbolometer sensors and their application in automotive safety. In J. Valldorf and W. Gessner, editors, *Advanced Microsystems for Automotive Applications 2003*, VDI-Buch, pages 137–157. Springer Berlin Heidelberg, 2003. ISBN 978-3-540-76988-0. doi:[10.1007/978-3-540-76988-0_13](https://doi.org/10.1007/978-3-540-76988-0_13).

- S. Yu and H.-S. Wong. Modeling the switching dynamics of programmable-metallization-cell (PMC) memory and its application as synapse device for a neuromorphic computation system. In *Electron Devices Meeting (IEDM), 2010 IEEE International*, pages 22.1.1–22.1.4, 2010. doi:[10.1109/IEDM.2010.5703410](https://doi.org/10.1109/IEDM.2010.5703410).
- S. Yu, B. Lee, and H.-S. P. Wong. Metal oxide resistive switching memory. In J. Wu, J. Cao, W.-Q. Han, A. Janotti, and H.-C. Kim, editors, *Functional Metal Oxide Nanostructures*, volume 149 of *Springer Series in Materials Science*, pages 303–335. Springer New York, 2012. ISBN 978-1-4419-9931-3. doi:[10.1007/978-1-4419-9931-3_13](https://doi.org/10.1007/978-1-4419-9931-3_13).
- K. Zaghloul and K. Boahen. Optic nerve signals in a neuromorphic chip: Part I and II. *Biomedical Engineering, IEEE Transactions on*, 51(4):667–675, 2004. doi:[10.1109/TBME.2003.821040](https://doi.org/10.1109/TBME.2003.821040).
- C. Zamarreño-Ramos, L. A. Camuñas-Mesa, J. A. Perez-Carrasco, T. Masquelier, T. Serrano-Gotarredona, and B. Linares-Barranco. On spike-timing-dependent-plasticity, memristive devices, and building a self-learning visual cortex. *Frontiers in Neuroscience*, 5(26), 2011. doi:[10.3389/fnins.2011.00026](https://doi.org/10.3389/fnins.2011.00026).
- W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Comput. Surv.*, 35(4):399–458, 2003. doi:[10.1145/954339.954342](https://doi.org/10.1145/954339.954342).
- V. V. Zhirnov, R. Meade, R. K. Cavin, and G. Sandhu. Scaling limits of resistive memories. *Nanotechnology*, 22(25):254027, 2011. doi:[10.1088/0957-4484/22/25/254027](https://doi.org/10.1088/0957-4484/22/25/254027).
- M. Ziegler, R. Soni, T. Patelczyk, M. Ignatov, T. Bartsch, P. Meuffels, and H. Kohlstedt. An electronic version of pavlov’s dog. *Advanced Functional Materials*, 22(13):2744–2749, 2012. doi:[10.1002/adfm.201200244](https://doi.org/10.1002/adfm.201200244).

Index

- électrolyte
 - liquide, 110
 - solide, 47
- étiquetage, 81, 102
- évolution génétique, 80

- AER, 61, 74, 77, 81, 138
- amorphisation, 39
- apprentissage, 19, 77
 - classification, 97, 140
 - couche par couche, 85
 - dynamique, 116
 - formalisation, 98
 - global, 82
 - Hebbien, 23, 47, 97, 109
 - non supervisé, 101
- auto-assemblage, 33

- bipolaire, 9
- Blue Brain, 147
- bottleneck de Von Neumann, 10, 18,
19, 48, 146
- BrainScaleS, 145

- CAM, 15
- CBRAM, 46
- cellule
 - bipolaire, 21
 - ganglionnaire, 21, 98
 - photosensible, 21
- chalcogénure, 38, 47
- champ récepteur, 24, 59, 88
- CMOL, 15
- codage
 - fréquentiel, 22, 64, 97
 - ordre, 22, 97
 - semi-fréquentiel, 100
- Cog Ex Machina, 145
- colorimétrie, 98
- concept cells, 24
- connaissance à priori, 17
- consommation, 131, 137, 139

- contrôle du gain, 36
- cortex visuel, 23
- couche
 - V1, 17, 23
 - V2, 24
- courants de fuite, 12
- crystallisation, 39
- croissance (PCM), 44
- crossbar, 11, 59, 82, 107, 128
 - passif, 12
- CRS, 13
- cumulativité, 10
 - CBRAM, 48
 - PCM, 40

- d-prime, voir sensibilité
- dépression, voir LTD
- détection
 - de bruit répétitif, 90
 - de piéton, 18
 - de trajectoire, 77
 - de véhicule, 81
 - de visage, 24
- dispositif
 - à jonction tunnel, 52
 - CBRAM, 46
 - contrôlé en courant, 8
 - contrôlé en tension, 8
 - défectueux, 12
 - memristif, 8
 - MRAM, 52
 - PCM, 38, 123
 - RRAM, 49
- donnée naturelle, 16
- drift (PCM), 131

- endurance, 132
- ERB, 91

- feedforward, 23, 75, 82, 120, 128
- FFT, 64
- file de priorité, 56

-
- filtre
 - banc de filtres, 91
 - Butterworth, 91
 - Gabor, 23, 98
 - passe-bande, 91
 - passe-bas, 92
 - fonction gaussienne, 81
 - FPNI, 15
 - HBP, 147
 - HMM, 90
 - implication matériel, 15
 - inhibition latérale, 64, 78
 - jitter, 90
 - Labyrinthe de Morris, 145
 - LFSR, 135
 - logique
 - opérateur, 63
 - reconfigurable, 12
 - loi log-normale, 134
 - loi normale, 33
 - LTD, 75, 134
 - fenêtre de, 107
 - probabilité, 98
 - LTP, 40, 75, 134
 - fenêtre de, 100, 107, 108, 124, 134
 - probabilité, 98
 - mémoire associative, 109
 - MAC, 19
 - macro-bloc, 59
 - macro-pixel, voir macro-bloc
 - marge de bruit, 13
 - memistor, 110
 - memristor, 8, 52
 - ferroélectrique, 53
 - MNIST, 102
 - modèle
 - asymétrique, 128, 131
 - cochlée, 91
 - compatible circuit (PCM), 42
 - itératif (NOMFET), 30
 - neurone, 58
 - phénoménologique (PCM), 40
 - poids synaptique, 98
 - MoNETA, 145
 - motif
 - AER, 81
 - auditif, 90
 - visuel, 74
 - MTJ, 52
 - multi-niveaux, 10
 - nœud, 56
 - Neurogrid, 147
 - neurone
 - implémentation, 115, 148
 - LIF, 58, 76
 - NOMFET, 28, 110
 - NP, 28
 - nucléation (PCM), 44
 - oxyde
 - binaire, 50
 - de pérovskite, 50
 - oxydoréduction (C₂BRAM), 47
 - période réfractaire, 78
 - PCM, 38, 120
 - pentacène, 28
 - perceptron, 20, 102
 - PMC, 46
 - poids synaptique
 - additif, 75
 - binaire, 99, 133
 - pseudo-multiplicatif, 75, 91
 - stabilité, 99
 - potentialisation, voir LTP
 - PRNG, 135
 - psychoacoustique, 94
 - psychovisuelle, 98
 - réseau convolutionnel, 74
 - résistance thermique, 43
 - rétine
 - biologique, 21, 74
 - impulsionnelle, 74
 - rétro-propagation du gradient, 23, 102
 - règle du Delta, 97
 - rafraichissement (PCM), 125
 - résistance drift (PCM), 42
 - ROI, 18
 - RRAM, 49
 - sélectivité, 94, 138
 - saillance, 98
 - sensibilité, 95
 - signal échantillonné, 93
 - simulateur
 - évènementiel, 56
 - à pas de temps, 64
 - Brian, 64
 - sneak paths, voir courants de fuite

- SpiNNaker, 146
- STDP, 22
 - asymétrique, 106, 118, 119
 - biologique, 23, 107
 - implémentation, 106, 121
 - limitations, 147
 - simplifiée, 75, 134
 - stochastique, 133
 - symétrique, 118
- stimulus
 - conditionné, 109
 - inconditionnel, 109
 - neutre, 109
- stochasticité, 10
 - CBRAM, 49
 - extrinsèque, 137
 - intrinsèque, 137
- synapse
 - 2-PCM, 123
 - binaire, 133
 - dynamique, 109
 - GABAergiques, 118
 - glutamatergique, 118
 - stochastique, 134
- SyNAPSE (DARPA), 144
- threshold switching effect (PCM), 42
- topologie, 59, 86, 128
- transcodage, 138
- transistor de sélection, 12, 47
- tungstène, 39
- unipolaire, 9
- variabilité, 103, 131
 - neuronale, 90
 - NOMFET, 33, 118
 - synaptique, 89
- wire bonding, 114
- WTA, 24
- Xnet, 56