



HAL
open science

Un calcul cortical pour les systèmes situés

Hervé Frezza-Buet

► **To cite this version:**

Hervé Frezza-Buet. Un calcul cortical pour les systèmes situés. Informatique. Université Nancy II, 2011. tel-00652806

HAL Id: tel-00652806

<https://theses.hal.science/tel-00652806>

Submitted on 16 Dec 2011

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

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

Un calcul cortical pour les systèmes situés

MÉMOIRE

présenté et soutenu publiquement le ...

pour l'obtention de l'

Habilitation de l'Université Nancy II
(Spécialité Informatique)

par

Hervé Frezza-Buet

Composition du jury

Rapporteurs : Philippe Tarroux, Professeur à l'École Normale Supérieure
Philippe Gaussier, Professeur à l'Université de Cergy-Pontoise
Peter F. Dominey, Directeur de Recherche au CNRS

Examineurs : Anne Boyer, Professeur à l'Université Nancy 2
Hélène Paugam-Moisy, Professeur à l'Université Lumière Lyon 2
Jean-Paul Haton, Professeur à l'Université Henri Poincaré, Nancy 1
Frédéric Alexandre, Directeur de Recherche à l'INRIA

Mis en page avec la classe thloria.

Remerciements

La rédaction de ce mémoire a été pour moi l'occasion de faire le point, c'est-à-dire de m'extraire de la « *course* » aux résultats et publications pour réfléchir et formaliser la démarche qui m'a animé dès le début de mon travail de thèse, démarche qui s'est prolongée depuis. Lorsque j'ai pris le temps de m'arrêter pour faire une synthèse, est venue immédiatement, au-delà de l'articulation d'une cohérence scientifique, une pensée pour les personnes dont la présence a été essentielle pour la réalisation de mon travail.

Commençons d'abord par le nerf de la guerre, l'argent, ce qui m'amène à mentionner Supélec qui m'emploie depuis plus de dix ans. Que l'on puisse s'impliquer dans des enseignements de qualité dans une école comme celle-ci est relativement attendu, et c'est effectivement une source quotidienne d'émulation intellectuelle. En revanche, ce qui peut paraître plus étonnant, c'est qu'une recherche très prospective comme celle que je présente ici ait pu s'y épanouir. L'image caricaturale d'une école orientée exclusivement vers la formation d'ingénieurs pour le milieu industriel en est quelque peu écornée, et en effet la réalité de l'inscription de Supélec dans le paysage scientifique est plus complexe, et plus riche, du fait notamment d'une volonté de cette institution de s'engager dans la recherche. Je me félicite de pouvoir contribuer à cet engagement, remerciant au passage mes collègues, au sens très large du terme, pour être les destinataires patients d'un humour quotidien toujours un peu limite. Au support institutionnel de Supélec, qui inclut une part déterminante de financements publiques, s'ajoutent bien évidemment des financements européens, nationaux et régionaux. Ces financements concernent les projets de recherche, mais également l'existence-même du milieu académique local avec lequel j'ai pu interagir, à travers le Loria principalement. Ces ressources sont le bien de tous. Elles font de nous, chercheurs, des citoyens privilégiés par nature, privilégiés de devoir pour faire notre métier être déchargés de contraintes commerciales. Que ceux qui pensent que ce propos est idéaliste mesurent l'ampleur du renoncement intellectuel auquel ils ont consenti.

Au-delà des moyens, il y a les personnes, et je ne pourrai bien sûr pas citer nommément toutes celles qui m'ont accompagné, que ce soient les collègues de Supélec, du Loria (avec une distinction particulière à la confrérie bidibulienne), ou d'autres institutions. Néanmoins, il m'apparaît incontournable ici de citer Frédéric Alexandre, qui a soutenu mon évolution en tant que chercheur bien au-delà de ma thèse. Il est à ma recherche ce que l'apparition spectrale d'un maître Shaolin est au jeune adepte du kung-fu tétanisé face à son premier adversaire. De façon plus anecdotique, je lui suis également redevable de l'effort d'avoir relu ce document malgré son volume, effort consenti également par Yann Boniface. Soyez-en tous deux remerciés. Je ne peux non plus omettre mes doctorants, Olivier Ménard, Lucian Alecu, et Bassem Khouzam, à qui je dois la plupart des contributions originales présentées dans ce document. À ceux-ci s'ajoutent indéniablement les stagiaires, pour la plus grande partie de Supélec et de l'Université Polytechnique de Bucarest, dont les apports logiciel ont été déterminants pour que mon travail puisse exister.

Enfin, en cette étape de synthèse d'une décennie de recherches, j'aimerais exprimer aux scientifiques qui composent mon jury mes remerciements. Le fait qu'ils consacrent de leur temps pour me permettre de présenter l'Habilitation à Diriger les Recherches, en prêtant attention à ce que j'ai pu proposer à la communauté, est pour moi un honneur.

Je souhaite, avant que nous entrions dans un propos plus technique, remercier ma famille pour accepter de faire de la place à mon travail de recherche, parfois envahissant, car leur soutien et leur adhésion me sont indispensables.

Aux miens.

Table des matières

Chapitre 1

Préambule

1.1	Définition de l'intelligence	2
1.2	Le pari de la pluridisciplinarité	4
1.2.1	Rôle de l'informatique dans une approche pluridisciplinaire de l'intelligence	6
1.2.2	Le contexte robotique	6
1.3	Ce qu'enseignent les « sciences molles » à l'informatique	8
1.3.1	L'approche située	9
1.3.2	La cognition est motrice	16
1.3.3	Repenser la modularité des systèmes	22
1.4	Précautions oratoires	23

Partie I Problématique de recherche

25

Chapitre 2

Vue d'ensemble

2.1	Enjeux pour l'informatique	27
2.1.1	Maîtrise de la complexité	28
2.1.2	Machines intelligentes	30
2.1.3	L'exigence de calculs spatialement répartis	31
2.2	Un projet de recherche informatique	32
2.2.1	Ambitions	33
2.2.2	Les ressources de calcul extensives comme objectif	34

Chapitre 3

Systemes Dynamiques

3.1	Formalisation	37
-----	-------------------------	----

3.1.1	Définitions	37
3.1.2	Attracteurs et points fixes	38
3.1.3	Stabilité	40
3.2	Modélisation par les systèmes dynamiques : encore un problème de robinet	40
3.3	Automates cellulaires	42
3.3.1	Là où le bât blesse	43
3.3.2	Mises à jour asynchrones	44
3.3.3	Automates à états continus	45
3.4	Modèles élémentaires	46
3.4.1	Dynamique du potentiel de membrane	46
3.4.2	Dynamique de population	47
3.4.3	Quid de la modélisation fine de la dynamique biologique ?	47
3.4.4	Modèles biologiques computationnels	48
3.5	Systèmes dynamiques comme modèle de mémoire temporelle	48
3.5.1	Principe	49
3.5.2	Modèles impulsionnels	50
3.5.3	Modèles fréquentiels	51
3.6	Systèmes dynamiques comme modèle de mémoire spatiale	52
3.6.1	Modèle de Hopfield	53
3.6.2	Champs de neurones dynamiques	53
3.6.3	Pertinence d’une étude empirique pour évaluer les champs de neurones	62
3.6.4	Bilan	63
3.7	Apprentissage et systèmes dynamiques	64
3.7.1	Techniques de <i>Machine Learning</i>	64
3.7.2	Techniques évolutionnistes	65
3.7.3	Apprentissage Hebbien	65
3.7.4	Apprentissage par renforcement	67
3.7.5	Conclusion	67
3.8	Simulation des systèmes dynamiques	68
3.9	Questions pour notre problématique	69
3.9.1	Que fait la dynamique ?	69
3.9.2	Pertinence du modèle impulsif	69

Chapitre 4

Le calcul cortical

4.1	Module cortical	73
4.1.1	Un pavage bidimensionnel d’unités	74

4.1.2	Quelle fonction pour un module ?	75
4.2	Architecture corticale	77
4.2.1	Indices primaires et secondaires	78
4.2.2	Multimodalité	85
4.2.3	La notion de <i>hiérarchie</i>	87
4.2.4	Discussion	89
4.3	Dynamique fonctionnelle du cortex dans son ensemble	91
4.3.1	Une organisation globale non supervisée	91
4.3.2	Un parallélisme fonctionnel cortical ?	94
4.4	Le <i>calcul cortical</i>	95
4.4.1	Propriétés « corticales »	95
4.4.2	Méthodologie de construction des modèles	97
4.4.3	Le modèle <i>grumpf</i>	97
4.5	Le modèle <i>bijama</i>	102
4.5.1	Organisation des calculs en « étages »	102
4.5.2	Connexions en bandes et étages « <i>corticaux</i> »	104
4.5.3	Contraintes de cohérence multimodale	105
4.5.4	Principe d' <i>auto-organisation</i>	107
4.5.5	Principe de l' <i>auto-organisation conjointe</i>	109
4.5.6	Le champ de neurone BINP	113
4.5.7	Extensions	114

Chapitre 5

Approche située et calcul cortical

5.1	Le cortex réalise une cognition motrice	118
5.2	Les temps	120
5.2.1	À quoi bon reconnaître les séquences ?	121
5.2.2	Le <i>temps physique</i> dans les systèmes neuro-mimétiques	122
5.2.3	Le <i>temps logique</i> dans les systèmes neuro-mimétiques	124
5.3	Récompense	124
5.4	Ajout de mécanismes de récompense dans <i>bijama</i>	125
5.4.1	Le problème du <i>balbutiement moteur</i> sur un cas simple	126
5.4.2	Architecture	128
5.4.3	Conclusion	131

Partie II Environnement **135**

Chapitre 6

Activités de recherche « collatérales »

6.1	Résolution d'équations aux dérivées partielles	138
6.2	GNG-T : Quantification vectorielle de données non stationnaires	140

Chapitre 7

Activités d'enseignement

7.1	Notre formation au métier d'enseignant	146
7.2	Enseignement à l'Université	146
7.2.1	Stages de DEA/Master informatique de l'UHP Nancy I	146
7.2.2	Cours au Master informatique de l'UHP Nancy I	146
7.3	Formation technologique	147
7.3.1	Algorithmique et programmation	147
7.3.2	Formation à l'apprentissage automatique	147
7.3.3	Projets et stages	147
7.4	Ouverture à la pluridisciplinarité	148
7.5	Formation Continue	148
7.6	Synthèse	149
7.6.1	Formation à la recherche	149
7.6.2	Enseignement	149

Chapitre 8

Dissémination de la recherche

8.1	Participation à des projets de recherche	154
8.1.1	Projet Européen CLASSiC	154
8.1.2	Projet CPER : InterCell	154
8.1.3	Projet Européen MirrorBot	156
8.1.4	Projet Avim du programme Robea	157
8.1.5	Projet « Télé-assistance mobile » du CPER-Lorraine	158
8.2	Actions d'animation de la recherche	158
8.2.1	Organisation de conférences	158
8.2.2	Autres actions	159
8.3	Publications	160

Chapitre 9

Production logicielle

9.1	Politique de développement logiciel	165
9.2	Visualisation 3D	167
9.3	Traitement d'images	167
9.4	Enseignement	168
9.5	<i>Middleware</i> réseau	168
9.6	La bibliothèque <i>grumpf</i>	170
9.6.1	Le noyau <i>grumpf</i>	170
9.6.2	Les outils de visualisation	171
9.6.3	<i>bijama</i>	171
9.6.4	<i>lgn2v1</i>	171
9.6.5	Le convertisseur <i>simpf</i>	173
9.7	La bibliothèque <i>booz</i>	173
9.7.1	Portage du calcul cellulaire interactif sur <i>cluster</i>	173
9.7.2	Le référencement paresseux de <i>booz</i>	173
9.7.3	Interactivité	174
9.7.4	Exemples de références	174
9.8	Logiciels pour l'apprentissage automatique	174
9.8.1	SVM et méthodes à noyaux	176
9.8.2	GNG-T	176
9.9	Plate-formes robotiques et <i>smartroom</i>	176

Partie III Perspectives

179

Chapitre 10

Projet de recherche

10.1	Cheminement	182
10.1.1	Bilan rétrospectif des travaux de thèse (1999)	182
10.1.2	Bilan des travaux à Supélec	183
10.1.3	Travaux en cours	183
10.1.4	Suite du projet de recherche	183
10.2	Fonctions cognitives	184
10.2.1	Champ d'application	184
10.2.2	Propriétés à explorer	185
10.3	Architectures informatiques	186

10.3.1 Architectures extensives	186
10.3.2 Inspiration d'autres grandes structures cérébrales 2D	187
10.3.3 Structures 3D	187
10.4 Systèmes dynamiques	187
10.4.1 Apprentissage lent	187
10.4.2 Apprentissage immédiat	188
10.4.3 Déséquilibre permanent ?	189
10.5 Supports de calcul	190
10.5.1 Modèles non-impulsionnels	190
10.5.2 Expansion physique des calculs	190
10.5.3 Suite de projet InterCell	190
10.5.4 Composants matériels génériques	191
10.6 Recherche collatérale	191
10.6.1 Résolutions d'équations différentielles	191
10.6.2 Applications de GNG-T	191
10.7 Synthèse	192
Bibliographie	193
Index	211

Chapitre 1

Préambule

Depuis les années 50 sont apparues, comme domaine d'étude, ce que l'on appelle les *sciences cognitives*, qui ont fédéré des disciplines telles que, entre autres, la neurobiologie, la psychologie, la philosophie, la linguistique... et bien entendu l'informatique. Sans revenir sur l'historique de la constitution des sciences cognitives, nous en retiendrons qu'elles sont apparues à partir du moment où les sciences sont devenues capables d'aborder la question de la conscience, en décortiquant et objectivant les phénomènes de mémoire, de perception, de langage et d'émotions. Objectiver le sujet, qui est au cœur des sciences cognitives, est l'expression d'un paradoxe dont nous parlerons dans ce chapitre, et la science informatique y a pris toute sa part.

L'Intelligence Artificielle (IA) a été le versant en informatique des sciences cognitives. Parler d'intelligence artificielle fait d'ailleurs toujours l'objet de polémiques, la question de savoir jusqu'où l'on peut dire qu'une machine est capable d'intelligence n'étant pas tranchée aujourd'hui. Face à cette indétermination, nous soutiendrons l'hypothèse dite de l'*IA forte*, qui propose de considérer que l'ensemble de ce que l'on peut observer chez l'homme en termes de langage, pensée, conscience, est strictement le fruit de son métabolisme, neuronal en particulier. Nous ne demanderons pas au lecteur d'adhérer à cette hypothèse, mais soulignons ici qu'elle motive les travaux et les orientations de recherches que nous présentons dans ce mémoire.

L'hypothèse d'IA forte trouve un écho particulier en informatique pour les raisons suivantes. L'informatique est une discipline qui dès l'origine [Turing, 1936; Church, 1936] a su abstraire la mécanique du traitement de l'information de son support. En effet, les machines de Turing ont existé bien avant d'être instanciées sur support physique. Selon cette perspective, qu'un traitement soit effectué sur silicium, au sein d'une clepsydre améliorée, ou sur un support neuronal ne change rien à l'affaire. La métaphore de la *chambre chinoise* de John Searle [Searle, 1980] illustre cette indépendance au support. Elle compare l'ordinateur à une personne enfermée dans une salle qui manipule des symboles chinois auxquels elle ne comprend rien, suivant pour ce faire un système de règles. Si l'on adhère à l'hypothèse d'IA forte, ce que ne fait pas Searle, le système de règles pourrait décrire une intelligence, équivalente à celle de l'Homme, dont l'opérateur est le moteur. Ce qui motive notre recherche informatique est l'hypothèse de l'existence de ces règles, autrement dit, d'un programme qui conduise à ce qu'une intelligence *de même nature* que celle de l'Homme puisse être instanciée par une machine de Turing.

Faire cette hypothèse, toutefois, ne permet pas de guider la conception du programme, ou plutôt d'un programme, qui puisse doter une machine d'intelligence. Il est alors nécessaire de trouver ailleurs les arguments permettant de concevoir ces programmes. Là encore, l'hypothèse d'IA forte joue un rôle central. Si on admet que l'intelligence dont l'Homme fait preuve n'est que le résultat de la *mécanique* de ses neurones, il devient pertinent de s'inspirer des neurones

pour concevoir un programme intelligent. L'argument ici n'est pas de dire que la voie neuro-mimétique est la seule qui puisse permettre de concevoir des machines intelligentes, mais plutôt de souligner que, sous l'hypothèse d'IA forte, cette voie permettra d'aboutir. En plus de cette garantie de succès, toute théorique car rien ne dit que la technologie permettra d'instancier ces machines, l'inspiration des neurones a un autre avantage pour l'informatique, qui se défend sans recourir à l'hypothèse d'IA forte. En effet, qu'ils suffisent ou non à rendre compte de l'intelligence, les mécanismes neuronaux qui s'expriment au sein du système nerveux sont un exemple de traitement de l'information de nature foncièrement différente de ceux conçus par les approches classiques de génie logiciel. L'étude de ces systèmes est alors susceptible d'apporter à l'informatique des paradigmes nouveaux, et c'est ce que nous défendons le plus ardemment dans nos travaux de recherche.

L'ensemble de ce chapitre consiste à préciser ces quelques lignes d'introduction, en abordant le talon d'Achille des sciences cognitives, à savoir la question de la définition de l'intelligence. Nous abordons ensuite la question de la pluridisciplinarité, saisie du point de vue de l'informatique, pour défendre l'apport des sciences dites « molles » à notre discipline. Nous nous trouvons en effet dans la situation où un recours aux sciences humaines et sociales peut être défendu comme une aide pour débloquer les difficultés que rencontrent des sciences pourtant « dures » lorsqu'elles abordent la question de l'intelligence.

1.1 Définition de l'intelligence

Dans le contexte des sciences cognitives, les chercheurs échangent sur les thèmes de la conscience, de la pensée, de l'intelligence, et il a été tentant pour les psychologues, comme pour les informaticiens, de définir cette intelligence. Sont afférentes à l'intelligence des notions de gestion d'événements nouveaux, de création de nouveauté, comme l'illustrent les quelques citations suivantes¹ :

« L'intelligence est la capacité à comprendre un contexte nouveau, et à réagir à cette nouvelle situation de façon adaptée » .

Richard Atkinson

« L'intelligence, ça n'est pas ce que l'on sait mais ce que l'on fait quand on ne sait pas » .

Jean Piaget

Ces définitions sont attaquables si l'on garde l'hypothèse de l'IA forte, car la seule nouveauté que l'on puisse créer en absence de la prédétermination d'une connaissance, d'un savoir faire, est une action purement aléatoire. D'autres tentatives de circonscrire par des mots l'intelligence se basent sur le fait de prendre l'Homme à témoin, en jouant sur le trait d'esprit et sur la question de l'art, épineuse elle aussi :

« L'intelligence, c'est ce qui permet d'entendre une musique là où d'autres n'entendent qu'un bruit. » .

Jean-Charles Terrassier

Nous pensons toutefois que ce type de pirouette, par laquelle certains auteurs se sortent de la difficulté de piéger l'intelligence dans les mots, est l'approche la plus constructive de l'appréhension scientifique de l'intelligence. Une des plus fameuses a sans aucun doute été élaborée par Alan Turing lui-même, lors de son célèbre test [Turing, 1950] : une machine réussit le test

1. Citations collectées le 27 juin 2007 sur le site <http://fr.wikipedia.org>, rubrique « Intelligence ».

lorsqu'elle arrive à se faire passer pour un humain auprès d'un humain... c'est à un humain, un vrai, qu'appartient la faculté de qualifier ou non une entité comme étant douée d'intelligence. La force de cette approche est qu'on ne demande pas à l'humain de justifier son choix, ce qui évite de retomber dans les problèmes inhérents à l'élaboration d'une définition. Ainsi Stevan Harnad distingue les systèmes explicites et les systèmes implicites² en ce que les premiers suivent une règle alors que les seconds se comportent *en accord* avec une règle [Harnad, 1990]. Dans ce dernier cas en effet, nul besoin de formuler la règle et là encore, même si l'auteur ne le dit pas, le fait de juger du comportement du système comme en adéquation avec une règle formulée relève de la responsabilité d'un être humain. Nous considérons que cette démarche, qui consiste à prendre l'Homme à témoin pour décider du caractère intelligent d'un élément de la réalité, est la plus pertinente non parce qu'elle évite de se poser la question d'établir une définition, ce en quoi notre position serait le recours à une solution de facilité, mais parce qu'elle soutient que la question de la définition de l'intelligence est fondamentalement mal posée.

Suivons le raisonnement du psychanalyste Jacques Lacan [Lacan, 1973] pour argumenter qu'il est légitime de se soustraire à la nécessité d'explicitier la notion d'intelligence. Sa démarche consiste à partir des paradoxes pour critiquer le *cogito* cartésien.

« Il est tout à fait faux de répondre à ce je mens que, si tu dis je mens, c'est que tu dis la vérité, et donc que tu ne mens pas, et ainsi de suite. Il est tout à fait clair que le je mens, malgré son paradoxe, est parfaitement valable. En effet, le je qui énonce, le je de l'énonciation, n'est pas le même que le je de l'énoncé, c'est-à-dire le shifter qui, dans l'énoncé, le désigne. Dès lors, du point où j'énonce, il m'est parfaitement possible de formuler de façon valable que le je – le je qui, à ce moment-là, formule l'énoncé – est en train de mentir [...] ou même, qu'en disant je mens, il affirme qu'il a l'intention de tromper.

[...] Disons que c'est de prendre sa place au niveau de l'énonciation qui donne sa certitude au cogito. Mais le statut du je pense est aussi réduit, aussi minimal, aussi ponctuel – et pourrait aussi bien être affecté de cette connotation du ça ne veut rien dire – que le je mens de tout à l'heure. Peut-être le je pense, réduit à cette ponctualité de ne s'assurer que du doute absolu concernant toute signification, la sienne y compris, a-t-il un statut encore plus fragile que celui où on a pu attaquer le je mens.

[...] Dès lors, j'oserai qualifier le je pense cartésien de participer, dans son effort de certitude, d'une sorte d'avortement [...] J'épinglerai la fonction du cogito cartésien du terme d'avorton ou d'homoncule » .

Jacques Lacan [Lacan, 1973]

Nous souhaitons défendre ici, par ce raisonnement, que circonscrire dans le langage l'intelligence, la pensée, la conscience, ou tout autre substantif pour qualifier ce qui nous apparaît si particulier à notre espèce, tombe sous le coup du paradoxe du *je mens* ou *je pense*, et ne fait qu'exprimer notre embarras face à l'expérience intime de notre conscience. Ce scepticisme face à l'expérience de la conscience est encore d'actualité, et il motive sans doute de nombreuses recherches en sciences cognitives.

« The new scepticism questions whether we even have the perceptual experience we think we have » .

[Noë, 2002]

2. Les premiers sont symboliques et les seconds connexionnistes dans son article, ce qui n'est pas notre propos ici.

Une réponse à ce scepticisme a été apportée par Varela (cf. [Duquaire, 2003] pour les références et une présentation générale), qui reproche aux approches symboliques comme connexionnistes des sciences cognitives de ne pas considérer le sujet. La cognition selon lui ne se définit que par des expériences *éprouvées* dans un corps. Varela propose, pour ce qui est des sciences cognitives, de faire des sciences « à la première personne », afin d'éviter de sombrer dans le nihilisme d'une rationalisation de la cognition. Il propose le recours aux pratiques bouddhistes d'attention/vigilances pour éprouver les modalités d'incarnation du sujet dans son corps.

Pour notre part, plutôt que d'articuler des définitions qui découleraient de ce scepticisme, ou de faire, comme Descartes, référence à Dieu, ou encore d'avoir recours à un discours à la première personne basé sur une expérience intime, nous préférons suivre la pensée de Lacan pour qui « *Il n'y a pas de méta-langage* » qui nous soit disponible pour dire cette expérience, ce qui évacue toutes ces questions. Nous ne chercherons donc pas à définir l'intelligence, la pensée ou la conscience dans ce document, et nous en parlerons malgré tout avec le langage comme s'il faisait fonction, pour ces questions, de méta-langage³, sachant pertinemment que cela ne tient pas debout. Il s'agit là d'une entorse à l'objectivité du discours scientifique, exigée par le contexte qui est le nôtre, que nous assumons d'autant plus qu'une de ses conséquences est de tordre définitivement le cou à l'homoncule.

Nous en resterons donc à la position de Turing, qui s'en remet à l'Homme pour éprouver l'existence d'une intelligence qui ne soit pas la sienne. Où mènent alors nos travaux, qui cherchent à trouver un système de règles qui puisse faire de la chambre chinoise une entité intelligente, à l'insu de l'opérateur ? Lacan soutient qu'il y a parole là où il n'y a pas de formule [Lacan, 1978, chapitre XIX], comme c'était le cas lorsque les planètes aux mouvements erratiques parlaient⁴ aux hommes, avant que Newton, par ses équations ne les réduisent définitivement au silence. Notre démarche, si elle devait aboutir, conduirait-elle, faisant de l'Homme un « *être parlé* » dans un langage informatique, à le réduire, par des formules, au silence ? C'est toute l'angoisse que suscite l'hypothèse d'IA forte auprès de ceux qui n'y adhèrent pas, alors que cette question n'est certainement, elle aussi, qu'un paradoxe de plus révélant l'incapacité de l'Homme à traiter l'expérience de la conscience par le langage.

1.2 Le pari de la pluridisciplinarité

La majeure partie des conceptions industrielles en informatique repose sur des approches modulaires, du type de celles que l'on peut modéliser via UML. Ces approches sont orientées par la spécification de modules et de leurs interactions, en y ajoutant parfois le raffinement de concepts d'héritage statique, et plus rarement de types génériques, dont la cohorte de programmeurs Java s'est longtemps passée, avant la version 1.5 de la plate-forme. Des produits comme la machine .NET de Microsoft, avec l'ajout de la notion de programmation objet dans VisualBasic et la promotion de C# ne dérogent pas à cette tendance. Disons pour résumer que l'informatique industrielle s'est engouffrée dans la programmation objet, tout du moins au sens des langages statiques comme C++ ou Java.

Restent néanmoins quelques approches à contre-courant de cette tendance, promouvant la programmation fonctionnelle et une algorithmique plus fine que les décompositions *top-down* de classes en sous-classes devenues traditionnelles, comme par exemple dans [Abelson *et al.*, 1984], où le type de programmation induite par l'usage de langages comme Caml, Lisp ou Haskell met l'accent sur la puissance de l'évaluation paresseuse. Ces approches sont toutefois minoritaires

3. C'est le parti pris qu'a choisi Lacan pour théoriser l'expérience analytique.

4. L'imparfait n'est malheureusement pas encore tout à fait de mise aujourd'hui.

dans l'informatique d'aujourd'hui, ce que l'on peut regretter, et le paradigme de modules à la « sauce UML » reste largement prédominant.

Ces paradigmes sans être propres à l'informatique, sont sans conteste portés par elle car elle en exploite la puissance d'expression afin de maîtriser la complexité des traitements. Notre discipline a, de ce point de vue, peu à apprendre d'autres disciplines. Il n'en reste pas moins vrai que l'informatique gagne à se nourrir d'autres approches du traitement de l'information, et c'est sur ce terrain-là que s'articulent nos recherches.

La pluridisciplinarité n'est pas nouvelle en informatique, certainement parce que le fait de disposer de machines de Turing universelles incite à modéliser nombre de phénomènes. Mentionnons par exemple les algorithmes génétiques [Goldberg, 1989] et les réseaux de neurones [Bishop, 2004], initialement issus de modélisations du vivant. Il est toutefois à noter qu'au fur et à mesure de leur maturation au sein de la discipline informatique, les liens avec le domaine d'inspiration se distendent. Aujourd'hui en effet, les algorithmes génétiques sont un des divers procédés de recherche opérationnelle, sans vocation à être un bon modèle de la génétique, et l'on constatera que les perceptrons multi-couches, technique reine des réseaux de neurones, ne sont décrits dans les livres que par un formalisme matriciel, vu comme une simple descente de gradient pour réaliser une régression aux moindres carrés.

Les domaines qui nous concernent n'ont pas encore atteint cette maturité, si bien que la question de la pluridisciplinarité reste vivace et nécessaire pour l'informatique. La question de l'objectivation de l'intelligence est aujourd'hui encore très ouverte, et requiert toujours une investigation conjointe des disciplines qui forment les sciences cognitives. Bien que l'argument n'ait rien de scientifique, nous souhaitons souligner que la nécessité d'une approche pluridisciplinaire, et l'ouverture qu'elle requiert vers d'autres disciplines, nourrit la motivation qui nous porte dans nos recherches.

Dans le cadre de nos activités, nous avons participé à des réflexions et travaux pluridisciplinaires, impliquant des biologistes, des médecins, des psychologues (cf. chapitre 8). Nous avons constaté que les chercheurs de ces disciplines sont, en général, de piètres informaticiens, portant sur notre discipline un regard parfois réducteur, voire proposant des modèles qui, d'un point de vue algorithmique, peuvent-être assez simplistes. Qu'est-ce à dire, si ce n'est que ce constat nous renvoie, en miroir, ce que sont nos propres compétences en biologie, psychologie, etc. alors même que nous côtoyons ces domaines depuis des années.

Nous souhaiterions soutenir ici qu'une des clés de la réussite d'une collaboration pluridisciplinaire est que chacun reste à sa place. Ceci signifie, pour ce qui concerne nos travaux, que nous n'apporterons jamais aux sciences cognitives qu'une compétence informatique, et qu'il appartient aux autres partenaires de voir dans nos travaux ce qui pourrait, éventuellement, nourrir leur discipline

Ceci précisé, un dialogue pluridisciplinaire ne doit pas pour autant s'interdire l'appropriation des concepts d'une discipline par une autre. Qu'un biologiste fasse le choix d'exprimer sa pensée, dans ce contexte, sous forme d'un algorithme est très constructif, même si l'algorithme, d'un point de vue informatique, est perfectible. C'est suivant cette conception que nous nous autoriserons à donner dans ce document notre point de vue sur le cerveau, le cortex en particulier, point de vue qui pourra paraître au biologiste assez caricatural, fait de jugements à l'emporte-pièce motivés par la nécessité de ramener la biologie du cortex à un modèle dont nous maîtrisons la complexité. Que le lecteur n'y voit pas l'affirmation d'une vérité qu'apporterait l'informatique sur la biologie, mais plutôt la présentation d'une démarche informatique qui, fière des disciplines

qui l'inspirent, leur rend hommage et s'énonce dans un discours qu'elles pourront recevoir pour en appuyer certains éléments et en contester d'autres. C'est tout le sens du dialogue que nous impose l'ambition des sciences cognitives.

1.2.1 Rôle de l'informatique dans une approche pluridisciplinaire de l'intelligence

Nous avons présenté dans [Frezza-Buet, 1999] le rôle qui nous semble appartenir à l'informatique dans le contexte des sciences cognitives, ce que nous reprenons succinctement ici. Un programme informatique est un objet ayant vocation à s'exécuter. Il répond à une syntaxe, celle du langage dans lequel il est programmé, à laquelle on sait de façon formelle et univoque associer une sémantique, qui décrit ce que doit exécuter la machine lorsqu'elle « *lit* » le programme. La sémantique du langage étant définie, un programme est une *spécification* d'une exécution. Le fait de ne disposer que d'objets formels qui s'exécutent est particulier à l'informatique. Les mathématiques savent en effet prouver l'existence d'objets sans les construire, la physique sait modéliser des propriétés⁵ sans que cela permette de les simuler. L'informatique, elle, fait nécessairement le lien entre une spécification formelle, le programme, et une exécution, c'est-à-dire une action *physique* dans le monde. C'est en ce sens que cette science peut jouer un rôle prépondérant dans une entreprise de modélisation. Les modèles informatiques, comme tous modèles, sont la formalisation par l'Homme d'une réalité qu'il expérimente, formalisation qui peut être une simplification, certes, mais qui a l'avantage d'être discutable, réfutable, et que l'on peut confronter à l'expérience. Ce qu'ont de plus les modèles informatiques est que, même si au regard de la réalité de l'expérience ils ne sont que des modèles, ils « *tournent* ».

Modéliser l'intelligence en s'inspirant de la « *mécanique* » du cerveau conduit à élaborer des modèles complexes, du point de vue du traitement de l'information. Là encore, l'informatique, au sens du génie logiciel cette fois-ci, possède les outils de gestion de cette complexité, plus que toute autre science. Autant nous avons soutenu au début de cette section 1.2 (page 4) que l'informatique a besoin de nouveaux paradigmes pour aborder la question de l'intelligence, autant il est clair que la mise en œuvre des dits paradigmes peut s'adosser aux techniques de génie logiciel disponibles aujourd'hui. Le gain pour les disciplines partenaires est clairement de reculer la limite de complexité en deçà de laquelle l'exécution concrète de modèles est envisageable.

1.2.2 Le contexte robotique

Nous défendons que l'informatique permet de définir des modèles de l'intelligence, et que ces modèles s'exécutent dans le monde physique. Une des modalités d'exécution est le contrôle, par ces modèles, de systèmes robotiques. La biologie et la psychologie enseignent que l'intelligence est incarnée dans un corps, et qu'elle ne se conçoit pas sans lui, ce que nous défendons au paragraphe 1.3.1. Donner un corps aux intelligences artificielles conduit alors à la question de la robotique, qui joue, dans le contexte des sciences cognitives, un rôle clé. C'est pourquoi notre laboratoire s'est doté, au cours des années, d'une plate-forme robotique d'expérimentation, présentée au paragraphe 9.9.

Dans le contexte des sciences cognitives en effet, la robotique opère comme une validation des concepts informatiques, certes, mais surtout des concepts biologiques. Dans la littérature en neurobiologie, les auteurs qui proposent des modèles montrent qu'il sont en adéquation avec les données issues d'une expérience. Or une expérience biologique consiste à isoler une situation dont on puisse contrôler les paramètres, et elle fait l'objet d'un a priori de l'expérimentateur,

5. Par exemple par des équations aux dérivées partielles.

qui la conçoit pour valider une hypothèse. Il est de fait étonnant de constater que les modèles biologiques, qui sont conformes aux expériences, soient si peu utilisables en robotique, alors que le système nerveux, objet de ces expériences, est le contrôleur le plus performant que l'on connaisse. C'est cette contestation-là qu'apporte l'expérimentation robotique, proposant un complément à la vérification expérimentale des modèles biologiques. Permettre à un biologiste de montrer qu'une hypothèse formulée dans son domaine a la propriété d'être opérante dans un contexte robotique est une forme de validation que seule une approche pluridisciplinaire peut autoriser. Nous considérons cette validation comme d'autant plus recevable que la robotique, si l'on exclut les automates, est un domaine où les sciences technologiques sont tenues en échec depuis qu'elles existent.

Ces difficultés ont d'ailleurs été perçues par les pouvoirs publics, au niveau national et européen, puisque les programmes de recherche robotique se tournent désormais vers les sciences cognitives pour trouver d'autres voies de progrès de cette discipline. L'Agence Nationale de la Recherche (ANR) lance un programme « *Systèmes Interactifs et Cognitifs* », qui stipule en préambule :

« La recherche et le développement en robotique, fortement interdisciplinaires, couvrent un large spectre de thèmes scientifiques. Ces recherches concernent les sciences de l'ingénieur, les sciences et technologies de l'information, les sciences du vivant mais aussi plusieurs aspects des sciences humaines. Un défi majeur concerne leur intégration au sein d'un même système : le robot. Il ne s'agit pas d'ajouter uniquement des techniques et des résultats de chaque discipline mais il est essentiel d'examiner chaque problème avec, dès le départ, une démarche pluridisciplinaire. »

Site <http://www.agence-nationale-recherche.fr>, appel à projet « *Systèmes Interactifs et Cognitifs* », consulté le 2 juillet 2007.

Au niveau européen également, le constat est posé du besoin de recourir à de nouveaux paradigmes de traitement de l'information pour la robotique, en considérant comme légitime l'inspiration des sciences du vivant pour proposer ces paradigmes.

« Artificial systems that [...] can achieve general goals in a largely unsupervised way, and persevere under adverse or uncertain conditions; adapt, within reasonable constraints, to changing service and performance requirements, without the need for external re-programming, re-configuring, or re-adjusting. [...] Work proposed should explore and validate the use of new, possibly bio-inspired, information-processing paradigms, and of models of natural cognition (including human mental and linguistic development), adaptation, self-organisation, and emergence; and take account of the role of systems embodiment and affordances. »

FP7 : Objective ICT-2007.2.1 (ICT-2007.2.2) : Cognitive Systems, Interaction, Robotics : extrait du fichier pdf de description complète du programme de travail, disponible sur le site http://cordis.europa.eu/fp7/ict/programme/overview2_en.html, consulté le 2 juillet 2007.

Pour résumer, les sciences cognitives ont besoin de la robotique comme plate-forme de validation, et la robotique a besoin de recourir aux sciences cognitives pour se sortir des impasses technologiques actuelles. Le rôle de l'automatique dans ce contexte nous paraît devoir être clarifié. Il est vrai qu'à l'heure actuelle, s'il s'agit de « *faire marcher* » un robot, nous recommanderions à l'ingénieur de se tourner vers une approche automatique. Toutefois, nous ne nous situons pas dans cette perspective applicative, du moins pas à court terme. Nous reconnaissons les apports de l'automatique en terme d'analyse des systèmes robotiques, mais cette discipline repose en

général sur la mise en équation du système à contrôler, qui permet par la suite de faire une recherche d'espace libre, une optimisation sous contrainte, ou toute autre forme de recherche opérationnelle. La force de l'automatique tient à la maîtrise des méthodologies de résolution mathématique et numérique des problèmes définis lors de la mise en équation, mais c'est aussi ce qui éloigne notre propos de cette discipline. Les rapports qu'entretiennent sciences cognitives et automatique avec la robotique sont très différents, même si les deux disciplines se heurtent, à leur façon, aux problèmes difficiles soulevés par la robotique autonome.

Parmi les défenseurs de la nécessité de recourir à des paradigmes différents en informatique, ainsi que de la nécessité de confronter l'intelligence artificielle à des tâches de contrôle robotique, rendons ici hommage aux travaux de Rodney Brooks [Brooks, 1990]. En effet, Brooks prétend que la conception de systèmes intelligents doit s'appuyer sur la nécessité de prendre en charge des boucles sensorimotrices. Nous reviendrons sur cette exigence, au paragraphe 1.3.1 consacré à l'*approche située*, mais retenons ici que Brooks pose l'expérience robotique comme un incontournable à la modélisation de l'intelligence, et ce à une période où l'IA était dominée par des techniques de manipulation symbolique désincarnées.

D'autres auteurs, comme Philippe Gaussier, partent du formalisme neuronal en proposant des architectures biologiquement inspirées modélisant l'arc perception-action [Gaussier *et al.*, 2000]. Ces architectures réactives ont l'énorme avantage de s'instancier sur des robots réels, placés dans des environnements intérieurs non simplifiés.

Mentionnons enfin des travaux plus proches des nôtres, réalisés au sein de l'équipe Cortex du Loria, à Nancy, où un modèle inspiré du cortex confère à un robot la capacité de résoudre des tâches d'attention visuelle [Rougier and Vitay, 2006]. Là encore, il s'agit de réalisations robotiques directement issues de la démarche des sciences cognitives.

1.3 Ce qu'enseignent les « sciences molles » à l'informatique

Nous avons soutenu au paragraphe 1.1 qu'il n'y avait pas de méta-langage pour parler d'intelligence. C'est à notre sens la raison pour laquelle la représentation du monde par des symboles est si difficile à atteindre, comme en témoigne le fameux *frame problem* rencontré par les approches symboliques [McCarthy and Hayes, 1969], qui illustre qu'un système d'inférence symbolique a besoin de connaître les effets d'une action, à savoir ce qu'elle change... mais aussi ce qu'elle ne change pas, ce qui constitue une infinité de choses. Discourir avec objectivité sur l'intelligence et le langage nous semble conduire inévitablement à ce type d'explosion combinatoire des symboles. En témoignent les travaux de linguistique qui se livrent à un véritable recensement des multiples usages de la langue, en s'efforçant de les regrouper en catégories⁶. Prendre l'Intelligence comme *objet* d'une démarche scientifique nous apparaît donc voué à l'échec, car porteur d'une inconsistance. Il n'empêche que la démarche scientifique classique a conduit aux avancées spectaculaires des sciences de l'ingénieur, dans lesquelles l'informatique et la robotique sont inscrites, et nous pensons qu'il est effectivement possible résoudre la question de l'intelligence par l'informatique, mais à condition de se garder d'entreprendre de formaliser ce type de concepts. Pour guider alors l'intuition du chercheur en informatique impliqué dans le projet ambitieux des sciences cognitives, il convient, pour un temps du moins, de s'abstraire de l'influence qu'exerce la culture des sciences formalisées, dites « *dures* », et de revenir à l'essence de comportement, qui relève du rapport de l'animal à son environnement, ramenant ainsi l'homme et son intelligence à sa

6. Ce point de vue de notre part s'est constitué suite aux exposés de leurs travaux de recherche par des linguistes, lors des diverses conférences organisées par le Réseau Grand Est des Sciences Cognitives (cf. paragraphe 8.2.2).

première nature. C'est pourquoi nous développons plus avant la question de l'approche située.

« [...] *it is unfair to claim that an elephant has no intelligence worth studying just because it does not play chess.* » .

Rodney A. Brooks, in [Brooks, 1990]

1.3.1 L'approche située

Les auteurs qui traitent de l'approche située situent la naissance de ce concept, en informatique, au niveau des travaux de Lucy Suchmann à la fin des années 80. Ces travaux portaient sur les interfaces homme-machine, et sur la pertinence de disposer d'un modèle (symbolique) de l'utilisateur (voir [Adelson, 2003] pour une vue d'ensemble). L'apport de cette approche est de ne pas considérer que l'homme déroule un plan lors de l'interaction, mais que son action est influencée en permanence par le contexte. Les plans sont, suivant cette perspective, des ressources permettant de connaître les caractéristiques du monde à exploiter pour atteindre un but, mais les actions, elles, sont le fruit d'une interaction constante avec l'environnement. L'exemple d'une descente en kayak illustre ce concept. Le plan dit qu'il faut passer à droite d'un rocher, puis entre deux autres, ... mais les actions à réaliser ne peuvent être déterminées à l'avance, le système doit, pour agir, rester couplé à l'environnement, en percevoir les modifications et rester réactif. Sans entrer plus loin dans ces travaux, nous retiendrons qu'ils font la promotion de systèmes qui ne peuvent se définir qu'en situation, ce qui a été à notre sens une avancée majeure en intelligence artificielle.

D'un point de vue méthodologique, un penseur comme Francisco Varela pousse les sciences cognitives dans de nouveaux retranchements, sous la pression de la nécessité de suivre une approche située en sciences cognitives (voir [Dennett, 1993; Duquaire, 2003] pour une meilleure présentation de la pensée de Varela que celle, succincte, que nous donnons ici). Il s'intéresse aux systèmes ayant la propriété d'*autopoïèse*, capables à l'instar des êtres vivants de maintenir leur organisation au sein d'un espace confiné, en se régénérant perpétuellement. Les relations de couplage de ces systèmes avec le monde sont si intimes que le système crée son environnement autant qu'il est façonné par lui, ce qui fait du système un agent capable d'*énaction*. La cognition ne se pense pas isolée du contexte dans lequel évolue le sujet.

Nous renvoyons à l'excellent historique établi par William Clancey pour une présentation des influences et de la progression des notions de cognition située, suivant des perspectives informatiques mais aussi biologiques et philosophiques [Clancey, 2006].

La question de la *représentation*

Dans leurs débuts, les sciences cognitives ont été dominées par l'idée que l'on pouvait, grâce à la logique et à un formalisme syntaxique, *représenter* le monde par des structures symboliques manipulables par des règles. Or dès leur application à des cas concrets, ces systèmes ont souffert de la difficulté d'*ancrer* les représentations symboliques dans le monde réel [Harnad, 1990]. Cet ancrage ne revient pas uniquement à mettre un nom sur les informations reçues du monde, car il est aussi nécessaire de pouvoir manipuler ces notions en accord avec les règles du monde. Haselager parle de *representational headaches* quand il s'agit d'interfacer les représentations avec la réalité. Il constate que les approches symboliques parviennent à rendre compte de situations de cognition anormales, comme les compétences des experts, mais échouent à rendre compte de situations quotidiennes. Ce sont ces constats qui ont motivé une remise en cause des approches classiques des sciences cognitives, ce qui conduit à considérer les approches situées. Cette remise en cause s'articule autour de cette question de la représentation.

« Nouvelle AI is based on the physical grounding hypothesis. This hypothesis states that to build a system that is intelligent it is necessary to have its representations grounded in the physical world. Our experience with this approach is that once this commitment is made, the need for traditional symbolic representations soon fades entirely. The key observation is that the world is its own best model. It is always exactly up to date. It always contains every detail there is to be known. The trick is to sense it appropriately and often enough » .

Rodney A. Brooks, in [Brooks, 1990]

Brooks invite à considérer l'intelligence artificielle sous l'angle d'une exploitation sensorimotrice directe du monde, un monde disponible qui n'a donc pas besoin d'être modélisé. Ceci est une réponse au *frame problem*, qu'appuie Andy Clark en soulignant que le cerveau recourt à des raisonnements globaux, du fait de sa structure connexionniste, et à des heuristiques bon marché, *fast and frugal heuristics*, qui sont très imparfaites en termes d'inférences logiques, mais qui permettent au sujet de réagir rapidement à une situation, de façon appropriée, même si cette réaction n'est pas optimale [Clark, 2003]. Il précise qu'utiliser les émotions pour conditionner les actions fait partie de ces heuristiques, ce qui renvoie au rôle des marqueurs somatiques introduits par Antonio Damasio [Damasio, 1995]. Ce rejet de représentations symboliques au profit d'une approche sensorimotrice de la cognition, en insistant sur son couplage avec le monde, est ce qui a présidé à l'entrée de la théorie des systèmes dynamiques dans les sciences cognitives et dans une *approche située* de la cognition (voir par exemple [Haselager, 1999]).

L'idée de l'approche située est de considérer comme système le triplet cerveau-corps-environnement et de ne plus avoir un point de vue « *isolationniste* » de la cognition [Clark, 1998a], se limitant au cerveau.

« An agent's physical body, the structure of its environment, and its social context can play as important role in the generation of its behavior as its brain » .

Randall D. Beer, [Beer, 2003]

« For it increasingly appears that the simple image of a general purpose perceptual system delivering input to a distinct and fully independent action system is biologically distortive. Instead, perceptual and action systems work together, in the context of specific tasks, so as promote adaptive success. Perception and action, on this view, form a deeply interanimated unity » .

Andy Clark, [Clark, 1998b]

La théorie des systèmes dynamiques, sur laquelle nous reviendrons au chapitre 3, permet de prendre en compte la dimension temporelle de la cognition [Beer, 2003] dans l'analyse du système cerveau-corps-environnement. Suivant cette perspective, on peut observer, comme c'est le cas pour les systèmes dynamiques, des comportements apparemment complexes ou dotés d'une composante cognitive, alors que les lois qui régissent le système sont bien plus simples.

« The trick is to organize and orchestrate some set of non-semantically specifiable properties or features so that a device thus built, in a suitable environment, can end up displaying « semantic good behavior » . The term semantic good behaviors covers, intentionally, a wide variety of things. It covers the capacity to carry out deductive inferences, to make good guesses, to behave appropriately upon receipt of an input or stimulus, and so on. Anything that (crudely put) looks like it knows what it is doing, is exhibiting semantic good behavior. » .

Andy Clark, [Clark, 2003]

Randall Beer s'est efforcé de montrer que l'on peut rendre compte de capacités dites cognitives par des systèmes dynamiques, sur des cas artificiels que l'on peut analyser. Il propose, à l'aide d'un réseau de neurones simple ayant des propriétés temporelles, la construction, par évolution, de systèmes réalisant une discrimination visuelle entre des ronds et des losanges [Beer, 2003], des modélisations d'expériences de psychologie [Beer, 2000], d'attention sélective [Slocum *et al.*, 2000], ainsi que l'articulation coordonnée des pattes de robots marcheurs [Beer, 2006]. D'autres approches à base de systèmes dynamiques ont également été étudiées dans le cadre d'aptitudes à l'imitation, avec le support de psychologues [Andry *et al.*, 2001]. Ce système dynamique est constitué de deux éléments, un agent professeur et un agent apprenant, dont les actions se couplent lors de la stabilisation de la dynamique.

Les systèmes dynamiques sont au cœur de l'approche située, que les auteurs fassent référence ou non à cette théorie, tant la question du couplage avec le monde, via des aptitudes sensorimotrices, est la clé du succès de cette approche, qui aborde la cognition par une refonte de la notion classique de représentation, même pour des tâches réputées « *gourmandes en représentation* »⁷. Ce que défend, entre autres, Haselager, c'est que la cognition est un *répertoire de comportements*. On n'embarque pas dans notre cerveau des informations sur le monde, comme le propose la notion de représentation, mais plutôt des schémas de comportement capables de résonner (et raisonner) avec l'environnement quand ils sont appropriés. L'approche située porte un coup sévère à la notion classique de représentation, et fait même de ce rejet un choix méthodologique d'approche de la cognition.

« *After all, the notion of representation was introduced to account for cases of cognition that behaviorism could not properly deal with. But if representations turn out to be omnipresent, what surplus value can they have for the study of cognition? [...] it may be a good research strategy to see just how far one can go without assuming representations from the start. Thus, the motto we will be defending is not "Away with representations", but rather "Don't use representations in explanation and modeling unless you really have to"»* .

[Haselager *et al.*, 2003]

Bien entendu, les tenants de l'approche classique n'ont pas manqué de formuler des objections, reprises par les auteurs favorables à l'approche située. En effet, Clark, tout comme Haselager, soulignent qu'il ne faut pas être trop prompt à qualifier de représentation toute forme d'intermédiaire entre les entrées et les sorties d'un contrôleur [Haselager *et al.*, 2003; Clark, 1998a]. Bien que ces intermédiaires permettent à un système de répondre différemment à une même entrée, parce que l'état interne n'est pas le même, on peut difficilement, même si les systèmes sont plus élaborés que de simples arcs réflexes stimulus-réponse, qualifier ces états de représentations. Ces états n'ont rien de langagier, comme souligné dans [Clark, 2003], et ne permettent pas vraiment de raisonnement hors-ligne, coupé du flux constant de stimuli, ce qui caractérise la mentalisation. On trouve des modélisations de calculs hors-ligne dans le cadre des systèmes dynamiques [Clark, 1998a; Clark and Grush, 1999], lorsqu'il s'agit de réaliser un contrôle moteur. On a en effet observé en biologie que, pour des questions de constantes de temps, il était nécessaire de se doter d'un simulateur pour disposer immédiatement des effets d'une commande afin de réaliser l'apprentissage, ce simulateur étant corrigé en fonction de l'effet réel de la commande. Ce type de simulateur, qui construit une prédiction, peut être vu comme un système hors-ligne, une mentalisation, mais on est loin de ce que l'on entend par mentalisation en langage courant, à savoir la capacité à articuler un raisonnement *dans sa tête*, soustrait à la pression des stimuli.

7. *representation hungry tasks* [Haselager *et al.*, 2003].

Que répondre à cela ? La notion de mentalisation conduit-elle nécessairement à conserver le recours au concept de représentation en sciences cognitives ? Nous soutenons que la notion de *mémoire provisoire*⁸ développée par Fuster [Fuster, 1997] est au cœur des processus de dissociation par rapport aux stimuli, et permet d'envisager la mentalisation sans avoir recours à la notion de représentation. Ce point sera détaillé au paragraphe 5.2.3, mais notons simplement ici que l'activité de réseaux récurrents, capable d'auto-entretenir leur activation, peut être vue comme une forme de mémoire qui ne requiert pas la présence de stimulus. Cet état peut subir une transition *on*, afin de stocker un contexte, puis subir une transition *off* lorsque ce contexte n'est plus pertinent. Ce type de mémoire agit comme une pile neuronale, pile étant à prendre au sens informatique du terme, et permet de ne pas perdre le fil d'un raisonnement articulé. Nous avons modélisé ce processus dans le passé, dans le cadre de la navigation d'un robot simulé [Frezza-Buet, 2002].

Enfin une représentation, qu'elle soit artistique, qu'elle soit un code, est fondamentalement destinée à être reçue, interprétée. Une représentation, c'est pour quelqu'un. On peut concevoir qu'une population de neurones soit une représentation d'un objet, d'un concept.... pour celui qui en observe l'activité à l'IRMf par exemple. En dehors de ce cadre, considérer qu'un cerveau manipule des représentations, cela revient à lui conférer un statut « *méta* » par rapport à elles, et à faire renaître ipso facto l'homoncule. Il nous semble que l'hypothèse d'IA forte exclut cette notion de représentation, qui véhicule le paradoxe que nous avons mentionné quant à la définition de l'intelligence. Nous sommes donc en phase avec l'orientation prise sur cette question par les tenants de l'approche située, et ne reconnaissons pas l'argument de la mentalisation comme contraire à cette perspective.

Reconnaissance de formes

La reconnaissance de formes est l'un des chevaux de bataille de l'informatique et du traitement du signal aujourd'hui, tant de nombreux problèmes industriels visent à automatiser les prises de décisions que nous, humains, réalisons naturellement (déclenchement d'alarme, détection d'éléments dans une scène, etc.). Les techniques d'apprentissage supervisé, réseaux de neurones, machines à noyaux, arbres de décision, sont aujourd'hui couramment employées. Ces techniques supposent que l'on peut catégoriser une entrée d'après ses caractéristiques propres, ou dit autrement, que l'information utile à la catégorisation est disponible dans le signal, charge aux informaticiens et traiteurs de signal de mettre en œuvre les procédures pour l'en extraire.

Cette conception guide une approche ascendante des systèmes, où l'entrée passe par une succession d'étages qui produisent, en sortie, la catégorisation. Ce que nous avons mentionné sur l'approche située, le couplage fort entre l'environnement et le système, remet en cause cette vue ascendante, ce que nous allons illustrer sur quelques exemples. Pour nous, il est clair que dans la plupart des situations réelles, *toute l'information n'est pas dans le signal*.

Prenons l'exemple des images de la figure 1.1. Ces images, une fois qu'on sait ce qu'elles représentent, sautent aux yeux, et l'on en vient à se demander comment on ne pouvait y voir que tâches informes. La réponse tient à ce que nous plaquons sur ces images un *a priori de forme*, afin de restituer une information qu'elles ne contiennent pas.

On pourra opposer à ce raisonnement qu'il s'agit d'un principe de complétion d'images, en exploitant les régularités qui se manifestent usuellement dans les images que nous regardons. C'est en effet ce qui est mis en œuvre sur la figure 1.2, où la complétion de l'image à l'endroit du texte est le résultat de l'itération d'une équation aux dérivées partielles [Aubert and Kornprobst,

8. Il s'agit, en quelques mots, d'une mémoire temporaire, destinée à stocker et mettre à jour le contexte de la tâche courante.

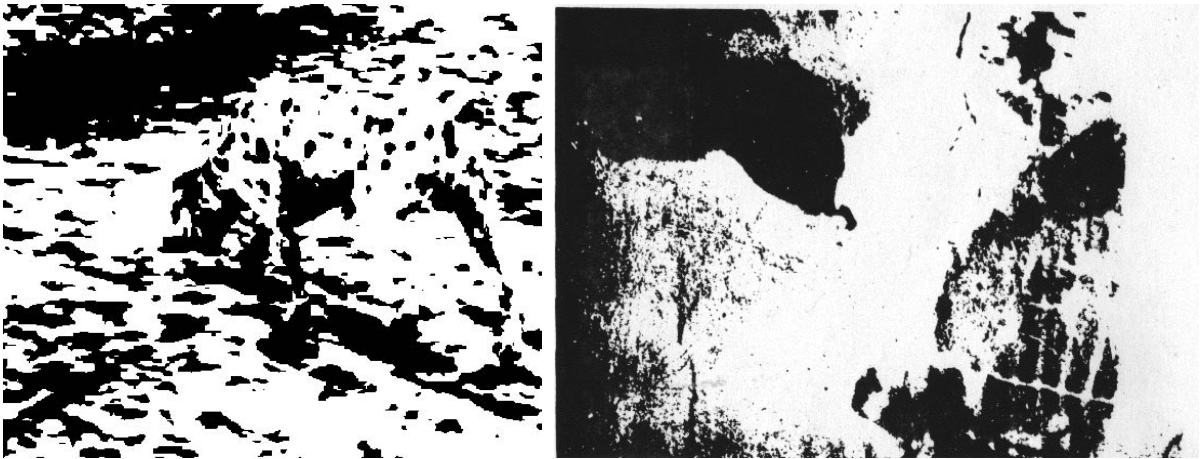


FIGURE 1.1 – Que représentent ces deux images ? Solution page 24. La figure de gauche est l'illusion de Richard Gregory, celle de droite l'illusion de Dallenbach.

2006, paragraphe 5.1]. Ce type de complétion a été aussi présenté comme modèle des processus visuels du cortex primaire [Mingolla *et al.*, 1999], mais la question que nous posons ici est de savoir si c'est ce qui est mis en œuvre dans le cas de la reconnaissance des images de la figure 1.1.



FIGURE 1.2 – Suppression de texte et complétion de l'image par des techniques à base d'équations aux dérivées partielles, tirées du site de l'équipe Odyssee <http://www-sop.inria.fr/odyssee/presentation>.

Le cas d'une image que l'on pourrait compléter est illustré sur l'image de gauche de la figure 1.3. On constate que, dans ce cas, l'on voit immédiatement, sans ambiguïté, ce que représente l'image, comme dans le cas de la figure 1.2 où le texte, même s'il est gênant, ne nuit pas à la compréhension de l'image. Ce qui est en jeu dans les images de la figure 1.1 est donc différent, et cela nous apparaît plus proche de l'ambiguïté de l'image de droite de la figure 1.3.

Le fait de trancher pour savoir ce que représente **B**, un B, le nombre 13, ou autre chose, ne peut pas être obtenu de la forme **B** elle-même, mais d'une action sur l'image de notre part. C'est ce type *d'action* qui nous permet de voir ce que représente les images de la figure 1.1. Nous prétendons qu'au regard de ces exemples, la course aux points supplémentaires de pourcentage de



FIGURE 1.3 – À gauche, image d’un dalmatien que l’on peut compléter sur la base des régularités observées dans les images en général. À droite, la même image, le caractère au centre, prend un sens différent suivant les contextes.

reconnaissance que l’on trouve dans la littérature de reconnaissance de formes (reconnaissance de caractères, reconnaissance de la parole, etc.) est un peu dérisoire.

Ces phénomènes de reconnaissance par une action sur la perception ne concernent pas que la modalité visuelle, on en trouve des exemples dans la modalité auditive pour la reconnaissance de la parole. L’action sur le signal auditif est guidée par le contexte sémantique, mais aussi par la modalité visuelle, par exemple via la lecture sur les lèvres. La célèbre expérience du « *ba-da-ga* » [McGurk and MacDonald, 1976] montre que lorsqu’on est soumis à un flux de syllabes prises parmi « *ba* », « *da* », « *ga* », que l’on entend, mais que l’on peut en même temps lire sur les lèvres, des erreurs de doublage du film volontairement introduites par les expérimentateurs peuvent nous conduire à entendre la syllabe articulée, alors qu’elle diffère de celle effectivement fournie sur la modalité auditive.

Ceci montre que la reconnaissance de forme est multimodale, contextuelle, et qu’il est vain de s’escrimer à faire de la reconnaissance en se basant sur un seul signal. Ceci milite pour une autre approche de la reconnaissance des formes, une approche en situation, ce qui conteste la suprématie actuelle des méthodes d’apprentissage supervisé, telles que les perceptrons multicouches ou les machines à vecteurs supports.

La causalité est nécessairement située

Nous avons montré que l’approche située est de nature à remettre en cause certains usages, et certaines conceptions, comme au paragraphe précédent en ce qui concerne la reconnaissance des formes. Nous souhaitons ici insister sur un autre concept, celui de causalité, qui lui aussi peut être éclairé par l’approche située. La question de la causalité est large, allant de la physique à la philosophie, et nous nous restreindrons à une approche opérationnelle, exploitable en sciences de l’ingénieur.

Dans ce domaine-là également, le sujet est vaste, et nous renvoyons le lecteur désireux d’approfondissement à l’ouvrage de Pearl [Pearl, 2000] qui aborde copieusement la question. Retenons ici que la détection de causalité est usuellement inspirée du critère de Wiener, établi en 1956, qui a été formalisé par Granger en 1969 (voir l’introduction de [Gourévitch *et al.*, 2006] pour plus de références). Le principe de ce critère est le suivant. Soient deux événements A et B , dont

on souhaite pouvoir dire si A est la cause de B . Soit I l'ensemble des informations autres que A dont on dispose, et $I \setminus B$ cet ensemble, privé de B . On dira que A est la cause de B , selon le test de Granger, si l'on peut mieux prédire A d'après I que d'après $I \setminus B$. Ce test est utilisé en traitement du signal [Gourévitch *et al.*, 2006], mais on en trouve aussi des raffinements en économie [Diks and Panchenko, 2006].

Parmi les approches de détection de causalité, nous citerons également les modèles graphiques utilisés pour les inférences bayésiennes, dont on trouvera une présentation dans [Bishop, 2006, chapitre 8]. Ces modèles visent à affecter un degré de plausibilité sur certains événements, au regard d'observations. Les événements ainsi considérés comme plausibles sont usuellement des causes retenues, parmi les différentes causes possibles qui conditionnent les observations. Certains auteurs critiquent le caractère statique de ces approches bayésiennes [Tian and Pearl, 2001], et s'inspirent d'études menées en économie pour montrer que la survenue de changements brusques est l'occasion de détecter des causalités. Lors d'un changement, leur hypothèse est que la probabilité de l'effet, sachant la cause, est inchangée, alors que celle de la cause sachant l'effet l'est. On peut ainsi savoir, de deux événements, qui cause l'autre, alors qu'une approche globale, basée sur la statistique des événements, noie les changements occasionnels dans les calculs de moyenne.

À notre connaissance, il apparaît que la détection de causalité se fonde sur des hypothèses exprimées par des probabilités conditionnelles, et il s'agit d'estimer la causalité à partir d'*observations* d'événements. En toute rigueur, si l'on revient à la définition bayésienne d'une probabilité conditionnelle, qui dit que la probabilité de A connaissant B est égale à la probabilité de l'événement joint A et B , relativisée par celle de B , ceci n'exprime aucunement une mécanique qui justifierait qu'un événement en produise un autre. Les approches bayésiennes ne sont que des inférences, des déductions, qui peuvent laisser à penser que des événements sont, du fait de l'observation de *corrélations*, en relation causale.

Comme nous l'avons fait dans [Frezza-Buet, 1999], revenons à un exemple caricatural, mais instructif et fermier. Tous les matins, quand l'on constate la simultanéité du chant du coq et du lever du soleil, comment savoir si c'est le coq qui fait se lever le soleil, l'inverse, ou même si ces deux événements sont provoqués par une cause supérieure, inaccessible aux sens ? Le fermier féru de science aura tôt fait de trancher la question, lui qui se lève avant l'aube, en forçant son coq à chanter au milieu de la nuit. Cela lui permettra de constater qu'effectivement, peu chaut au soleil de ses activités nocturnes.

Autrement dit, nous défendrons ici que réaliser une expérience est un moyen d'accéder à la causalité, ce qui suppose de pouvoir agir sur le monde.

La causalité prend alors une autre ampleur que dans le cadre de l'inférence à partir d'observations, car elle est un des outils dont un être agissant peut jouer pour exploiter la physique de son environnement. En fait, pour un agent situé, ce qui compte n'est pas la vraie causalité physique, mais simplement de disposer d'une règle permettant, par ses actions, de conduire le monde dans un état désiré. La causalité, suivant cette perspective, est une notion corollaire de l'approche située, et, dans la mesure où elle est abordée par les sciences cognitives, fait la promotion d'une cognition conçue dans le système cerveau-corps-environnement.

L'approche située, dont nous avons constaté l'empreinte qu'elle laisse dans la conception de la cognition, tout comme dans celles de notions classiques de reconnaissance de formes et d'inférence causale, amène à considérer l'action comme pivot des processus cognitifs. C'est l'objet du paragraphe suivant.

1.3.2 La cognition est motrice

Nous avons présenté, dans les paragraphes précédents, l'approche située comme étant l'un des apports, majeur selon nous, des « *sciences molles* » à l'informatique. Nous défendrons ici, en continuité avec la promotion d'une approche située, que non seulement il convient de considérer les systèmes comme indissociables de leur environnement, mais qu'en plus, *tous* les processus cognitifs sont de nature motrice. Cette affirmation radicale peut paraître intégriste et réductrice, et tous les auteurs défendant l'approche située ne vont pas jusque là [Clark, 2002]. Nous-mêmes, informaticiens, aurions peine à amener des preuves de nature biologique ou psychologique, mais ce que nous soutiendrons dans l'ensemble de ce texte, avec la plus grande force, c'est qu'adopter cette position intégriste en informatique est constructif, tant cette position remet en cause les conceptions classiques de la discipline. Nous pourrions toujours en revenir quand notre discipline en aura fait le tour, mais l'heure est bien d'avantage à l'innovation en ce sens qu'à dresser un bilan, aujourd'hui. Nous partirons, pour articuler l'intégrisme moteur qui sera le notre, du point de vue d'Alain Berthoz :

« On ne peut rien comprendre au fonctionnement du cerveau si l'on ne sait pas que son problème principal est de mettre en mouvement des masses » .

Alain Berthoz, [Berthoz, 1997]

Nous abordons cette question par les arguments les plus immédiats, à savoir les architectures sensorimotrices qui exhibent des comportements qui apparaissent plus élaborés que de simples réflexes. Nous répondrons ensuite à l'objection qui vient naturellement à l'esprit, mettant en doute le caractère moteur de la perception. Nous aborderons les questions du raisonnement et du langage, en montrant que même si l'on ne bouge pas, à l'exception des articulateurs lorsqu'on parle effectivement, il s'agit bel et bien d'une activité motrice, comme nous le suggère l'étude du cortex frontal et son analyse par Joaquin Fuster [Fuster, 1997].

Les architectures sensorimotrices

« In fact, these old distinctions (between perception, cognition and action) may sometimes obscure, rather than illuminate, the true flow of effect. In a certain sense, the brain is revealed not as (primarily) the engine of reason or quiet deliberation, but as the organ of environmentally-situated control. Action, not truth and deductive inference, are the key organizing concepts. » .

Andy Clark, [Clark, 1998b]

Andy Clark prend l'exemple d'une mouche [Clark, 2003] et décrit les arcs réflexes suivants. Dès que les pattes d'une mouche ne touchent plus le sol, le réflexe de battement d'ailes est déclenché. Un autre arc réflexe déclenche un saut de la mouche si une ombre diffuse se dessine au dessus d'elle. La combinaison de ces réflexes conduira à un comportement adapté de la mouche. L'idée est que des réflexes sensorimoteurs, par leur exécution, génèrent les conditions qui président à l'exécution d'autres réflexes élémentaires, etc. On peut imaginer des situations plus complexes, où l'exécution d'un arc moteur conduit à l'inhibition d'un autre, où les perception/action de ces boucles ne sont pas directement en prise avec le monde extérieur, mais reçoivent en entrée un état interne et agissent en modifiant un autre état interne. Dans un tel système, il n'y a pas de superviseur, mais simplement une population de boucles sensorimotrices qui s'influencent les unes les autres. C'est cette idée qui a guidé la conception de la *subsumption architecture* de Brooks, qui met justement en œuvre les activations interdépendantes de boucles sensorimotrices pour le contrôle de robots en environnement réel [Brooks, 1994; Brooks and Stein, 1994; Bergener et al., 1999].

D'autres approches, ayant le même souci d'asseoir la cognition robotique sur les boucles sensorimotrices, s'appuient plus explicitement sur la modélisation de structures biologiques. Parmi elles citons les travaux de Phillippe Gaussier qui permettent à des robots Koala⁹ de se localiser et de retourner à leur « nid » [Gaussier *et al.*, 2000], utilisant des architectures neuronales modélisant les *cellules de lieux*¹⁰ entre autres. Au dessus de la même architecture sensorimotrice, d'autres capacités cognitives ont été modélisées, comme l'utilisation de cartes cognitives, à base de cellules de lieu également [Gaussier *et al.*, 2001] et l'apprentissage de séquences comportementales par imitation [Gaussier *et al.*, 1998]

Nous avons nous-même proposé une architecture sur robot simulé, inspirée de la notion de propagation d'appel au sein du cortex [Burnod, 1989] qui apprend, à partir de détections de séquences d'observation, à invoquer les commandes motrices adéquates en fonction de l'objectif motivationnel courant [Frezza-Buet and Alexandre, 2002].

Ces modèles sont des modèles réactifs, s'appuyant sur l'usage et l'apprentissage de boucles sensorimotrices, pour lesquels on peut reconnaître toutefois des aptitudes cognitives. Leur force tient à ce qu'ils sont en contact étroit avec leur environnement, comme préconisé par l'approche située. Rappelons ce que dit Brooks de l'environnement (cf. page 10) : « *The trick is to sense it appropriately and often enough* ». C'est sur cette base que nous allons montrer en quoi la perception est également une composante motrice de la cognition.

La perception est motrice

Il est courant de penser que la perception, c'est une question de capteur, c'est recevoir de l'information. Nous avons déjà mis à mal cette vue par la promotion de l'approche située, mais nous souhaitons dans ce paragraphe insister sur ce point, car il nous apparaît que l'ignorer conduit à rester dans les impasses actuelles de l'intelligence artificielle. Parler de perception motrice, si on regarde de plus près la psychologie et la biologie humaine, n'est en rien un oxymore ; ça saute aux yeux.

Nous argumenterons cette idée principalement dans le cas très étudié de la vision, qui ne se ramène pas si intuitivement à un concept moteur d'un premier abord. Commençons par cette célèbre phrase de Merleau-Ponty :

« *La vision est palpation par le regard* » .

Maurice Merleau-Ponty [Merleau-Ponty, 1964]

La notion de palpation suppose une action, visant à acquérir, à piocher une succession d'informations sur le monde. Ce principe a été appliqué à la robotique par Brooks, et Andy Clark donne d'autres exemples [Clark, 1998b]. D'un point de vue phénoménologique, la question de savoir si la perception du monde est une *Grande Illusion* fait débat. En effet, si l'on se ramène à l'anatomie de la rétine [Hubel, 1994], la vision centrale n'est précise et colorée que sur quelques degrés d'angle solide, et l'image qui s'y projette est soumise à de brusques glissements, du fait des saccades que nous faisons fréquemment. La grande illusion, c'est que malgré cela, nous avons une *expérience* d'un monde stable, détaillé en tous points et pas uniquement là où l'on fixe.

Cette stabilité a été expliquée par certains auteurs, comme cela est rapporté dans [Clark, 2002; O'Regan and Noë, 2001] qui contestent toutefois ces explications. En effet, il est contestable que nous rassemblons les informations échantillonnées par les saccades en une structure interne cohérente et stable. Cette vision représentationnaliste est en contradiction avec l'approche

9. Robots de type char, de 30cm×30cm, <http://www.k-team.com>

10. Il s'agit de neurones observés dans l'hippocampe du rat, qui s'activent lorsque le rat est à un endroit donné de son environnement.

que nous défendons, et une autre explication de cette stabilité apparente tient justement à remettre l'action, ou plus précisément l'exploitation de contingences sensorimotrices, au cœur de l'analyse phénoménologique de la perception, de la vision en l'occurrence. Cette approche est abondamment argumentée dans [O'Regan and Noë, 2001; Noë, 2002; Myin and O'Regan, 2006], auxquels nous renvoyons le lecteur pour des références plus précises relatives à l'argumentation et aux travaux mentionnés dans ce qui suit.

Prenons un exemple de perception simple : le mystère de la lumière du réfrigérateur [O'Regan and Noë, 2001]. Pour savoir si cette lumière est allumée ou éteinte, il nous faut ouvrir la porte, et constater qu'elle est allumée. Si un doute nous saisit, ou qu'une tâche requiert de s'assurer de l'état de cette lumière, nous irons à nouveau ouvrir la porte du réfrigérateur. Nul besoin de garder une représentation de la lumière (sa couleur, son intensité), il suffit de savoir qu'ouvrir la porte permettra de répondre aux questions sur le lumière en cas de besoin. La perception de cette lumière est donc indissociable d'une action, d'une palpation. Elle suppose de plus la propriété essentielle, comme mis en avant dans [Clark, 2002], de *s'articuler dans le temps*, ce qui est une propriété fondamentalement motrice, comme nous le verrons au paragraphe suivant concernant le cortex frontal.

Reconsidérons la question de la grande illusion d'un monde visuel stable et détaillé sous l'éclairage... du frigo. Quand on touche une table du bout des doigts [Myin and O'Regan, 2006], on en perçoit des portions discontinues, mais nul besoin de mécanisme de remplissage pour restituer, dans le cerveau, la présence du plan de la table entre les doigts. Si on avait un doute, on pourrait toujours vérifier en déplaçant les doigts. Il en va de même pour le point aveugle de la rétine, les objets qu'il s'y projettent, même s'il ne sont pas reçus par le cerveau, ne sont pas considérés comme manquants (voir figure 1.4). Ce qui est notable sur cette figure, c'est que lorsque le vélo disparaît, la texture de fond ne paraît pas discontinue.

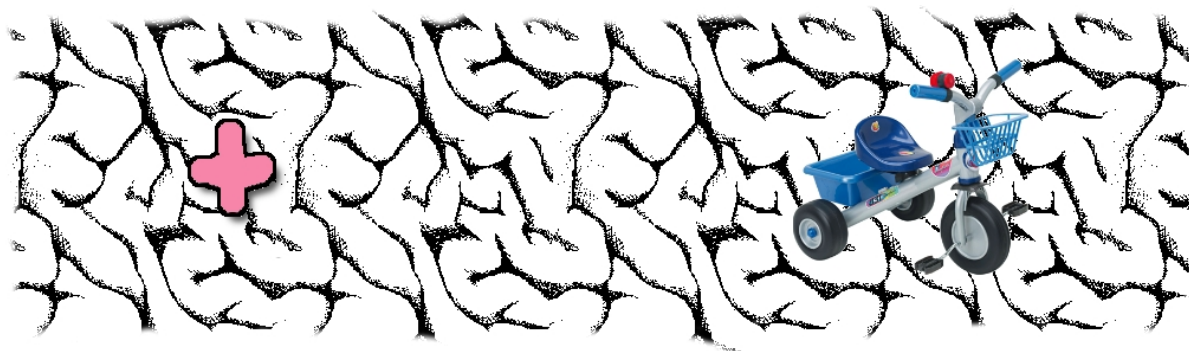


FIGURE 1.4 – Expérience du point aveugle. Fermez l'œil gauche, et fixez la croix avec l'œil droit. Trouvez la distance à la feuille pour laquelle le petit vélo disparaît.

Poursuivons le raisonnement tel qu'il est articulé, dans [O'Regan and Noë, 2001] par exemple. Ce qui crée la perception, ce sont les contingences sensorimotrices, le fait de savoir qu'en ouvrant la porte du frigo, j'accède à la lumière. Pour ce qui est de la vision, ces contingences sont bien plus complexes. Certes, si l'on est intéressé par un détail sur la droite du champ visuel, une saccade vers ce détail est une action qui permettra de l'analyser, mais ce n'est pas tout. Si un objet est perçu occulté, comme par exemple une chaise dont le dossier dépasse derrière une table, on sait comment se déplacer dans le monde 3D pour modifier l'occultation. Pour des raisons *géométriques*, il y a un lien entre la distance qui nous sépare des objet et la façon dont leur aspect est modifié quand nous déplaçons notre point d'observation. Cette distance qui le

sépare de l'objet est appréhendée par le sujet : personne n'a l'impression qu'une chaise stimule sa rétine, nous la percevons là où elle est... dans le monde.

Les auteurs définissent alors ce qu'est une perception visuelle, comme une contingence sensorimotrice influencée par les occultations et les changements de position [Myin and O'Regan, 2006]. À partir du moment où on reçoit une information du monde par un système revenant à projeter des lignes droites (trajet de la lumière) d'un point initial (le point de vue) vers des objets, ces lignes s'arrêtant à l'intersection la plus proche, on a une perception qui a toutes les caractéristiques de la vision, quant aux occultations d'objets, à leur modification lors de nos actions de déplacement, etc. Un télémètre laser est sujet aux mêmes contingences, par exemple. Ces régularités sont indépendantes de la forme effective du capteur, elles existent pour des raisons purement dépendantes de la géométrie du monde, et cette géométrie préside nécessairement aux contingences sensorimotrices que nous exploitons, c'est-à-dire les effets de nos actions sur l'état de nos capteurs.

Ainsi, il est possible de voir avec la peau... ce qu'ont montré les expériences de Bach-y-Rita dans les années 70 (toujours d'après [O'Regan and Noë, 2001]). Des sujets aveugles ont porté un dispositif composé de 400 vibreurs organisés en deux matrices 20×10 . Ces matrices sont déposées sur le dos des sujets, de part et d'autre de la colonne vertébrale. Les sujets tiennent dans leur main une caméra qu'ils peuvent orienter à leur convenance (ce point est essentiel), et dont l'image est traduite grossièrement sur la matrice de vibreurs, qui reste de faible résolution. La relation entre les mouvements de la main qui tient la caméra et la réaction des vibreurs est identique à ce que nous avons décrit, elle est sensible aux occultations, relative à des lignes qui partent d'un point (le foyer de la caméra) vers les objets, etc. Cette stimulation cutanée a donc toutes les propriétés de la vision, en terme de contingences sensorimotrices.

Les sujets rapportent qu'ils finissent par ne plus percevoir de vibration dans leur dos, mais qu'ils ont une perception *distale* des objets. Les objets sont éprouvés dans le monde, à leur place, et non sur le dos. Si l'on active un zoom de caméra à l'insu des sujets, ceux-ci se protègent le visage avec les mains, et *reculent*, alors que le zoom a lieu dans leur dos. Ce n'est donc pas le capteur, l'œil qui fait la vision, mais bien la nature des contingences sensorimotrices exploitées par le sujet pour palper son environnement, et les sujets de Bach-y-Rita voient, et le terme est approprié, avec la peau de leur dos. Cet exemple explique pourquoi il est naïf de croire que le cerveau redresse l'image qui se forme sur la rétine, il n'en a nul besoin. Des expériences ont même montré qu'un sujet, qui porte des lunettes inversant le haut et le bas finit par voir à l'endroit, et qu'il est même capable, s'il ne les porte que la moitié du temps, de s'adapter immédiatement s'il met ses lunettes. De même, un sujet qui porte des lunettes dont les verres sont bicolores, avec la moitié gauche bleue et la moitié droite jaune, ne perçoivent plus de changement de couleur lié à l'endroit où est dirigé leur œil. Confrontés à du gris pour tester leur adaptation, ils en perçoivent la couleur différemment *selon l'orientation de leur œil*, alors que *la stimulation physique rétinienne est la même*.

« The enactive, sensorimotor account explains how it can be that we enjoy an experience of worldly detail which is not represented in our brains. The detail is present – the perceptual world is present – in the sense that we have a special kind of access to the detail, an access controlled by patterns of sensorimotor dependence with which we are familiar. The visual world is not a grand illusion » .

[Noë, 2002]

Il n'y a donc pas de grande illusion : imaginer une chaise est une chose, mais cela ne la rend pas présente. Si en revanche on n'en perçoit un peu le dossier derrière une table, et que ce bout de chaise se comporte suivant les contingences sensorimotrices que nous connaissons, la chaise

toute entière nous sera présente et l'on parle de *perception amodale* [Noë, 2002] quand on perçoit ce qui n'est pas directement saisi par les capteurs. Il y a bien perception de la chaise, même si l'idée que nous nous en faisons est complètement fautive, il suffira d'aller vérifier, par une action, s'il s'avérait que la forme de la chaise avait une importance. De plus, nous ajouterons ici que considérer la stabilité du monde comme une illusion, c'est encore une fois mettre l'expérience du sujet en premier plan, la faire rentrer dans le cadre du discours scientifique et, selon nous, s'embourber dans le paradoxe du cogito (cf. page 3).

De nombreux travaux montrent à quel point il est possible de se jouer, dans le cadre de la vision oculaire, des limitations de la perception du sujet. Il est possible, sous certaines conditions, de modifier une scène visuelle massivement sans que le sujet s'en aperçoive, ce que Daniel Simons appelle le *Change Blindness*. Nous renvoyons le lecteur à [O'Regan and Noë, 2001] pour plus de référence à ces expériences, ainsi qu'aux exemples du site de Kevin O'Regan¹¹ et à ceux du site de Daniel Simons¹².

Nous rajouterons à cette analyse de la perception comme éminemment active, que cette propriété motrice se manifeste aussi selon nous dans des processus cérébraux qui n'ont rien de moteur, en ce sens qu'ils ne déclenchent pas de mouvement. Nous considérons en effet que les *feedbacks* au sein des circuits neuronaux sont une forme d'action, interne, visant à filtrer activement le signal. Ces *feedbacks* existent pour la vision [Hupe et al., 1998], ce qui a été ignoré par nombre de modèles *bottom-up* dans les années 80 et 90 [Thorpe et al., 01], à l'exception notable des travaux de Grossberg [Grossberg, 2003]. En effet, bien que la rétine effectue des traitements *feedforward* [Hubel, 1994] de type filtrage, ces traitements sont relayés au niveau des corps genouillés latéraux, pour être soumis aux actions *feedback* du cortex [Grossberg, 2003].

Il nous apparaît, au regard de ces analyses, qu'il est indispensable de considérer la perception comme un processus de manipulation active sur le monde, cohérent avec les couplages forts défendus dans le cadre de l'approche située. Percevoir n'est pas recevoir le monde, mais s'en saisir avec appétence, là où il nous propose des prises.

Ce caractère de la perception est appelé *grabiness* dans [Myin and O'Regan, 2006]. Considérer que le monde *offre* des prises, aux modalités multiples et combinées, pour une perception qui en est la saisie, nous apparaît donner un fondement à la théorie écologique de la perception de Gibson [Gibson, 1979] qui avance le concept d'*affordance* du monde, par opposition à une perception passive. De plus, cette vision de la perception et plus généralement de la cognition comme d'un couplage moteur avec l'environnement est en plein accord avec les descriptions du rôle du cortex frontal chez l'homme, tel que décrit par Joaquin Fuster. Ce lien vers une structure anatomique corticale est, pour nous qui sommes informaticien et qui nous inspirons du tissu cortical dans nos travaux, la passerelle qui devrait permettre d'aborder les notions de cognition motrice et située dans le contexte de la modélisation informatique, ce qui est l'ambition de nos recherches.

Le cortex frontal

Nous ne soutiendrons pas ici que l'Homme est l'animal qui trône au sommet du palmarès de l'Évolution, tant il reste probable que les insectes nous survivront en cas de pollution majeure.

11. <http://nivea.psych.univ-paris5.fr> consulté le 18 juillet 2007.

12. http://viscog.beckman.uiuc.edu/djs_lab/demos.html consulté le 18 juillet 2007.

Toutefois, il y a bien, chez l'Homme, une particularité, une niche écologique au sommet de laquelle il se trouve, et cette niche nous paraît être celle de la capacité à *sérialiser*. En effet, baser sa survie sur le comportement semble être ce que notre espèce a optimisé, et se comporter implique réaliser une suite d'actions, à partir des sollicitations fournies par le monde. Or on ne fait qu'une ou peu de choses à la fois, et cette contrainte oblige à dérouler sur la *ligne* du temps des motivations et des objectifs aux variables et conditions multiples. Faire une chose à la fois, et baser sa survie sur la complexité de ce que l'on peut réaliser sous cette contrainte, est le propre d'un comportement élaboré. Toutefois, nous recevons des stimulations sur notre rétine, notre peau, notre cochlée, notre système vestibulaire, ... en nombre colossal, et à chaque instant c'est une foule de données parallèles qui arrivent au cerveau via nos capteurs. Réduire ce flux en un chapelet d'actions semble être notre spécificité, et lorsque nous déroulons un comportement complexe au fil du temps, il s'agit de ne pas le perdre, ce fil.

Nous renvoyons le lecteur à l'ouvrage de Joaquin Fuster [Fuster, 1997, chapitre 8] pour une analyse détaillée du rôle fonctionnel du cortex frontal. Joaquin Fuster défend que la partie du cerveau qui permet de générer et suivre le fil du comportement est le cortex frontal¹³, qui a la particularité chez l'Homme d'être le plus développé du règne animal puisqu'il occupe 29% de son cortex. Il s'agit pour cette structure de mettre en adéquation la sérialisation du comportement avec des exigences hédoniques, des récompenses. Elle est en cela le pôle cortical de l'action organisée. Elle est le siège d'anticipations, mais aussi de stockage de *mémoire provisoire*, que nous avons déjà mentionnée page 12, qui permet de garder le contexte d'une action « *en tête* ». Par exemple, si l'on décide de sortir au restaurant, il convient de se diriger au préalable dans le placard afin de remplacer le survêtement que l'on porte par une tenue plus appropriée. Or dans la placard, notre regard peut être attiré par une tenue d'apiculteur, et il serait mal venu de l'enfiler et d'aller à la ruche récolter du miel. C'est l'un des symptômes des lésions frontales, la distractibilité, et le sujet atteint peut subir des troubles au niveau de la planification à long terme, comme des troubles de l'action immédiate (mettre de la mousse à raser sur sa brosse à dent), mais aussi, ce qui est à mettre en rapport avec le paragraphe précédent, des troubles de la gestion des saccades oculaires. Sérialiser suppose, et nous informaticiens le savons bien, de hiérarchiser, et nos comportements s'articulent en buts et sous-buts, charge au cortex frontal de jouer le rôle de pile pour conserver le but courant, qui a échappé à l'apiculteur de notre exemple.

Là où, à notre avis, l'analyse de Fuster est remarquable, c'est qu'elle souligne qu'un raisonnement, c'est quelque chose d'articulé, qui ordonnance une foule d'idées sollicitantes, en un discours prédicatif et hiérarchisé de pensées. En ce sens le raisonnement est quelque chose de moteur, même si tout se passe dans la tête sans le moindre mouvement. Le langage articulé est lui aussi moteur, et pour des raisons bien plus intimes que celles dues au fait qu'il mobilise des articulateurs, cela tient à sa structure. Chez l'Homme, Fuster écrit que le cortex frontal applique des compétences motrices (la sérialisation) à des activations cérébrales autres que le contrôle des muscles, d'où le caractère sériel et articulé de nos raisonnements. Nous considérons cette analyse comme en mesure d'unifier la cognition sous la bannière de la sérialisation motrice, et la prévalence des contingences sensorimotrices que nous avons soutenues dans le paragraphe précédent s'y inscrit tout naturellement.

« *The organism moves its eyes, repositions its body in order to get a better perceptual grip on the objects that surround it, and thereby attempts to advance in the execution of the hierarchy of ongoing projects it is engaged in* » .

[Myin and O'Regan, 2006]

13. Nous ne ferons pas ici de distinction entre préfrontal, frontal. Nous appelons cortex frontal la partie du cortex à l'avant du sillon central.

Suivant la perspective de l'approche située, et à la lumière du rôle de sérialisation du cortex frontal et de ce qu'il implique sur la nature du langage, *la notion de sémantique se conçoit comme le fait que le langage est un comportement situé* émanant d'êtres animaux, soumis à leurs exigences biologiques et aux mécanismes hédoniques qui tirent le comportement des individus vers la satisfaction de ces exigences.

Bonne nouvelle

Nous avons mis en avant dans les paragraphes précédents les arguments, auxquels nous souscrivons, qui plaident en faveur d'une cognition fondamentalement motrice, dans toutes ses composantes. Même si les modélisations des mécanismes neuronaux du cortex frontal sont précieuses, on ne sait pas encore aujourd'hui dériver de ces points de vues des algorithmes d'intelligence artificielle. Il y a toutefois une bonne nouvelle face à ces difficultés, c'est qu'aborder la cognition par la voie de robots situés et agissants, même s'ils sont aujourd'hui balbutiants, est une voie qui ne sera pas limitée à la simulation de crocodiles, êtres aux capacités sensorimotrices déjà remarquables en comparaison avec ce que savent faire nos robots. En travaillant sur des comportements situés, même simples, il semblerait, d'après les études biologiques et psychologiques que nous avons mentionnées, que l'on touche déjà du doigt la nature intime des raisonnements les plus abstraits, alors qu'aucune observation, à notre connaissance, n'est venue en aide aux chercheurs qui empruntent la voie du représentationnalisme et de la logique formelle pour émuler l'intelligence. Nous ne prétendons pas ici que le soutien de la biologie à l'approche située nous rapproche plus de la solution, l'avenir le dira, mais il est clair que ce soutien est une aubaine, dont les travaux de recherches que nous menons ont vocation à tirer le plus grand parti possible.

1.3.3 Repenser la modularité des systèmes

Dans cette partie du document, nous étudions les apports des sciences « *molles* » à l'informatique, et avons identifié l'approche située et la promotion d'une cognition motrice comme des éléments permettant de guider la conception de systèmes intelligents. Nous souhaitons ici mentionner, pour conclure, un apport qui porte plus au cœur de notre discipline, concernant la notion de modules organisés en système.

Les sciences de l'ingénieur, l'industrie en général, reposent sur une organisation en systèmes de modules. Un module a vocation à ce qu'on lui *délègue* une tâche, dûment spécifiée. Dans cette notion de délégation, il y a l'exigence forte que celui qui délègue doit pouvoir ignorer comment le module s'y prend pour réaliser la tâche, ce qui permet au commun des mortels de pouvoir allumer la télévision et changer de chaîne sans avoir la moindre notion d'électronique. Cette conception des systèmes en modules *opaques* fait la force de nos technologies et est centrale en informatique. Le module communique avec les autres par des interfaces d'entrées sortie isolées dans le module, dont la forme, du fait de l'opacité, est en général indépendante de la façon dont va opérer le module.

L'approche de problèmes complexes par une conception de systèmes modulaires, au sens précédemment décrit, est une culture très ancrée en informatique, et très efficace aussi, mais force est de constater que le système nerveux, système efficace lui aussi, n'est organisé de cette façon à la rigueur qu'au niveau du neurone, même s'il est probable qu'un biologiste contesterait cette assertion. En effet, les interactions entre groupes de neurones sont en général massives, au point même que l'élément d'un groupe voit une grande partie d'un autre groupe, et vice versa. Quand on isole une structure du cerveau, que ce soit sur des bases anatomiques ou fonctionnelles, il est clair qu'elle ne constitue pas une structure opaque pour les autres structures auxquelles elle

est reliée, tant elle reçoit en son sein d'afférences provenant de neurones qui lui sont extérieurs, et tant sont nombreux les neurones en son sein qui exportent de l'information vers d'autres structures.

Pourtant, et nous limiterons cette argumentation au cortex, il semble que certaines zones aient un rôle fonctionnel précis. Il y a dans l'organisation du cortex cette ambiguïté, de notre point de vue, qui tient à ce que le tissu cortical est un substrat à la surface duquel s'organise des modules, modules difficilement isolables du reste tant la connectivité expose tout le module à l'extérieur. Le cortex est-il une soupe de neurones massivement connectés, ou un pavage de modules aux rôles fonctionnels définis? Il nous apparaît que la souplesse du tissu cortical, en terme de réorganisation et de capacité à traiter des informations de modalités diverses, tient à ce que la réponse à cette question est certainement « *on peut très bien être les deux* », ce qui est une idée plutôt neuve quand on la rapatrie dans la discipline informatique.

De ce point de vue, l'inspiration de la biologie du cortex est un questionnement profond de la notion de modularité, et devient alors une problématique très informatique, puisqu'il s'agit bien de comprendre comment organiser l'information dans une population d'éléments traitant du signal, les neurones. On montrera au paragraphe 4.2.3 qu'en plus de l'opacité des modules, c'est aussi la structuration hiérarchique de l'information pour le traitement de tâches complexes qui est contestée par la structure du cortex, tant les modules corticaux sont soumis à des *feedback* les uns sur les autres, et tant le comportement est le fruit de l'ensemble de l'activité du cerveau, et non d'une décision que l'on peut isoler à un endroit donné, qui serait la sortie du système.

Il nous semble clair que la façon dont les sciences de l'ingénieur abordent la modularité est le fruit d'un raisonnement prédictif et articulé, et que les modules suivant lesquels nous organisons nos technologies sont le reflet de ce que nos cerveaux savent faire, à savoir le raisonnement sériel que nous avons mentionné au paragraphe précédent. Il nous apparaît qu'à observer les structures neuronales qui produisent ce raisonnement, nous soyons conduits, nous informaticiens, à sortir des ornières des paradigmes habituels de la modularité. Le cerveau a cet avantage, cette nouveauté pour l'informatique, qu'il n'a pas été produit par un cerveau, et l'appropriation des modalités de traitement de l'information de ce système biologique est à notre sens un axe de progression à ne pas négliger dans notre discipline.

1.4 Précautions oratoires

Nous nous sommes déjà, au paragraphe 1.1, autorisé à parler d'intelligence, de conscience, sans avoir à définir ces termes. Nous souhaitons ici prendre deux autres précautions oratoires, l'une à l'intention du lecteur biologiste, l'autre à celle de ceux qui considèrent comme vain le fait de vouloir reproduire la cognition par une machine de Turing.

Premièrement, nous parlerons dans ce document de biologie, et en particulier du cortex. Nous n'avons jamais observé de cortex ailleurs qu'au restaurant, accompagné de petits légumes, et il est clair qu'il ne nous appartient pas, en tant qu'informaticien, de dire ce qu'est le cortex. Comme précisé au paragraphe 1.2 page 5, les mentions que nous ferons à la biologie sont un regard d'informaticien, assumé comme tel, sur ce que nous en avons compris. Quant aux algorithmes que nous proposons, le vocabulaire biologique utilisé pour les décrire est là pour imaginer une présentation technique, parfois aride sans cet artifice. Ce vocabulaire a aussi vocation à rendre hommage aux concepts biologiques qui ont inspirés notre travail. Toutefois, il reste incontestable qu'il appartient aux biologistes de reconnaître dans les systèmes que nous proposons l'accord éventuel des mécanismes qui les composent avec quelque réalité biologique. C'est tout le sens d'une collaboration pluridisciplinaire réussie.

Deuxièmement, on pourra ne retenir de ce document, surtout si l'on adhère pas à la pertinence du contexte pluridisciplinaire que nous y défendons, que la simple présentation d'un certain type de calcul, basé sur des unités de traitement numériques à grain fin, ainsi qu'une méthodologique permettant d'en canaliser l'utilisation. Bien que nous considérons cette approche comme minimaliste, et regrettable, il est vrai qu'en toute rigueur, ce que peut proposer un informaticien, c'est un algorithme, et c'est effectivement ce qui est produit, en tout état de cause, par nos recherches.

Solution des jeux page 13

La figure de gauche représente un dalmatien en train de renifler le sol, la tête est à gauche. La figure de droite représente un gros plan sur une vache qui nous fait face.

Première partie

Problématique de recherche

Chapitre 2

Vue d'ensemble

Nous proposons dans ce chapitre de situer la problématique de recherche qui nous anime, en soulignant le fait que cette problématique appartient au domaine de l'informatique, contrairement à ce que pourrait laisser penser les nombreuses références que nous ferons à la neurobiologie dans ce document. Notre propos est de tracer les contours d'un projet de recherche qui s'est précisé durant environ les cinq dernières années écoulées, au sein de Supélec, et qui s'est nourri en particulier des collaborations continues qu'il entretient avec l'équipe de recherche Cortex du Loria, à Nancy, dirigée par Frédéric Alexandre.

Ce projet souhaite promouvoir l'intérêt pour le traitement de l'information pris comme objet d'étude, d'une inspiration de l'organisation anatomique et fonctionnelle du cortex cérébral, humain en particulier. Nous avons soutenu au chapitre précédent qu'il nous paraît du ressort des neurobiologistes de voir dans nos travaux matière à faire progresser leur propre appréhension de l'objet d'étude qui est le leur, le cerveau. Aussi ne tenterons-nous pas de défendre ce point. Nous espérons toutefois que notre participation à des manifestations pluridisciplinaires ainsi que nos publications permettront d'exposer nos travaux aux différentes perspectives qui s'intersectent au point des sciences cognitives, afin que les chercheurs non-informaticiens de ces disciplines puissent s'exprimer sur nos travaux.

Nous articulons par conséquent ici la définition et les enjeux d'un projet de recherche informatique que nous souhaitons porter aussi longtemps qu'il permettra d'apporter des idées neuves dans cette discipline.

2.1 Enjeux pour l'informatique

Nous nous proposons de définir dans un premier temps notre projet de recherche par les questions qu'il traite, ces questions étant à notre sens des enjeux de l'informatique d'aujourd'hui. Ces questions s'articulent autour de trois pôles.

Le premier est celui de la maîtrise de la complexité, question chère à notre discipline, qui motive entre autres les progrès des techniques de génie logiciel, mais aussi l'existence d'un nombre croissant de *middleware* permettant d'offrir des briques élémentaires de plus en plus abstraites et élaborées pour la conception de logiciels. La maîtrise de la complexité dépasse d'ailleurs le strict cadre de l'informatique et touche tout ce qui peut être décrit sous la forme d'un système.

Le second pôle est celui de la conception de machines intelligentes situées, pour lesquelles on ne possède pas vraiment de technologie aujourd'hui. Nous avons évoqué au chapitre précédent la portée de cette question dans le contexte des sciences cognitives, mais elle a également une portée économique puisque, depuis l'avènement de la robotique industrielle, les machines ne sont plus

des outils manipulés par l'homme mais ont vocation à le remplacer. La Communauté Européenne par exemple considère qu'une évolution de la robotique vers plus d'autonomie pourrait permettre de ré-industrialiser l'Europe. Pour ce qui est de l'informatique, la question qui se pose est celle de la calculabilité de l'intelligence, et cette question est encore très ouverte (et aussi mal posée, comme nous l'avons vu au chapitre précédent).

Enfin, le troisième pôle autour duquel s'articulent les questions abordées par notre projet est plus orienté vers les architectures des machines, puisque notre recherche fait la promotion de calculs spatialement répartis. Ce point est relatif à l'incarnation de nos modèles sur un support physique, ce que nous ne pouvons éluder puisque nous défendons la nécessité de *situer* les traitements informatiques.

2.1.1 Maîtrise de la complexité

Nous avons déjà évoqué au paragraphe 1.2 que nos travaux de recherche abordent la question de la modularité des systèmes, question qui justement est au cœur de la maîtrise de la complexité de nos technologies. En effet, pour ce qui est des systèmes construits par l'homme, ils peuvent se voir comme un agencement de modules qui communiquent via des interfaces normalisées, ou du moins spécifiées, comme c'est le cas pour les modules informatiques. Ainsi, un objet manufacturé complexe, comme une voiture, est un assemblage de modules, eux-mêmes composés d'autres modules, qui forment un tout cohérent du fait que chacun des modules répond à un *cahier des charges* qui, à un moment du processus de conception, a été *explicité* par un humain. Bien que les problèmes relatifs à la gestion de cette complexité, comme par exemple la preuve de programme, méritent encore un effort de recherche conséquent, il est clair que l'approche de la conception de systèmes modulaires que nous avons décrite est devenue aujourd'hui l'approche « *classique* » des sciences de l'ingénieur.

À côté des systèmes manufacturés, l'on a affaire à d'autres systèmes, constitués eux aussi d'éléments en interactions, mais qui n'ont pas la structure modulaire dont notre technologie porte l'empreinte. C'est le cas des systèmes économiques, dont la stabilité est discutable, mais aussi des écosystèmes, parfois fragiles, ou des organismes vivants dont les propriétés d'adaptation, de régénération et d'organisation ne cessent d'émerveiller qui les étudie. Ces systèmes sont incontestablement complexes, au sens où nous l'entendons, mais les mécanismes qui ont présidé à leur élaboration semblent assez éloignés d'une décomposition rationnelle de leur fonctionnement en une hiérarchie de sous-modules dont on pourrait exhiber une spécification.

Illustrons cette idée par l'exemple du cortex cérébral, sachant que les arguments que nous avançons valent pour la plupart des systèmes organisés naturels¹⁴. Le cortex est une structure régulière, et d'un certain point de vue assez simple. En effet, il y a relativement peu de gènes qui codent son élaboration, si bien que cette structure est relativement homogène. La complexité que nous lui prêtons semble donc être le fruit de sa taille, environ dix milliards de neurones. Pousser à l'extrême ce raisonnement conduirait à dire que le cortex « *n'est que* » la formidable juxtaposition, en masse, de circuits élémentaires. Cette affirmation est sans doute simpliste, mais elle n'est pas dénuée de fondement car le cortex est effectivement un pavage bidimensionnel de colonnes corticales [Mountcastle, 1957; Ballard, 1986; Burnod, 1989] et des zones aux fonctions diverses ont néanmoins une organisation neuronale similaire [Miller *et al.*, 2001]. Nous détaillerons ces points au chapitre 4.

Ramenée dans le contexte de l'informatique, cette affirmation soutient qu'il existe des systèmes aux propriétés de calcul complexes qui soient la juxtaposition d'un nombre élevé de

14. Naturels au sens où ils ne sont pas fabriqués par l'homme.

système élémentaires simples. Un système ayant cette nature est simple à spécifier : il suffit premièrement de spécifier le composant élémentaire, deuxièmement la nature des interactions entre un composant et ses voisins, et troisièmement le nombre de composants. Le premier point est un problème de spécification classique, le second également, mais sa complexité reste limitée si l'on considère un système à la connectivité homogène, et le troisième, lui, se résume à peu de choses près à un entier, alors que c'est là que réside la complexité du système global.

Ce type de système repose sur la possibilité d'une *émergence* de propriétés globales lors de la mise en connexion d'un grand nombre de composants élémentaires, qui ne possèdent pas ces propriétés. Comme rapporté dans [Deguet *et al.*, 2006] auquel nous renvoyons le lecteur pour une synthèse des questions relatives à la définition de l'émergence, certains auteurs considèrent que c'est la propriété d'émergence qui fait la complexité de ce qu'on appelle les *systèmes complexes*. Cette notion d'émergence reste toutefois difficile à formaliser, recourant parfois à des critères subjectifs, comme la surprise du concepteur face au fonctionnement du système qu'il a produit. Est même proposé dans ce contexte un test d'émergence qui, à l'instar du test de Turing pour l'intelligence (cf. paragraphe 1.1 page 3) repose sur un humain pour qualifier ou non une propriété comme étant émergente. D'autres approches, plus objectives, se réfèrent à la théorie de l'information et à la complexité des descriptions. L'idée est que le système est décrit par un langage L_1 , et ses propriétés par un langage L_2 , et qu'il y a émergence quand L_2 ne contient pas de raccourci pour décrire le système défini en utilisant L_1 . Dit autrement, *le moyen le plus simple de prédire le fonctionnement du système est la simulation*. Cette définition est elle aussi discutable, car elle n'implique pas que quelque chose de radicalement différent de ses règles de fonctionnement soit produit par le système, ce qui est une caractéristique exigée parfois pour attester l'émergence. Cette notion de « *radicalement différent* » est approchée par les travaux de formalisation d'Aleš Kubík qui reposent sur des *systèmes de grammaires* et comparent les langages qu'ils génèrent [Kubík, 2003]. Kubík considère une bande bidimensionnelle, où plusieurs grammaires peuvent opérer en parallèle des réécritures. Les configurations de bandes possibles suite aux réécritures, à partir d'un état initial, forment le langage généré. Kubík montre que certaines configurations du jeu de la vie (comme les *gliders*) permettent de générer un langage qui ne se ramène pas à la superposition¹⁵ des langages générés par chaque grammaire (i.e. chaque cellule), et sont selon sa définition émergentes. D'autres configurations initiales en revanche n'ont pas cette propriété, alors que l'automate cellulaire est le même. Cette approche est ce que nous avons trouvé de plus proche pour aborder l'émergence dans nos travaux, puisqu'elle se base sur des agents/grammaires correspondant à des unités de calcul en chaque point d'une surface (la bande 2D). Elle ne requière pas, de plus, de savoir exprimer une propriété dont on souhaite montrer qu'elle émerge, puisque l'émergence est attestée par la comparaison entre le langage généré par le système complexe, et la superposition des langages générés par chacun de ses composants. Il suffit de montrer que ces langages sont différents, ce qui est toutefois difficile dans notre cas.

Miser sur des systèmes émergents est à double tranchant. Tout d'abord, il semble difficile de trouver un composant élémentaire dont la participation à un système où il est répliqué en masse conduise à quelque chose d'intéressant, comme le suggère l'étude de Stephen Wolfram sur le cas simple des automates cellulaires unidimensionnels, où l'auteur constate la rareté des automates pouvant conduire à des comportements non triviaux, desquels on peut attendre une capacité de calcul intéressante [Wolfram, 1984; Gutowitz and Langton, 1988]. Cette difficulté est le quotidien des chercheurs impliqués dans la conception de systèmes neuro-mimétiques, dont nous sommes. Deuxièmement, il est mathématiquement difficile d'établir un lien formel entre la propriété globale que possède le système et la propriété de ses éléments, voire il est difficile

15. Cette notion a un sens précis non détaillé ici.

de formaliser la propriété globale. Rappelons ici la difficulté de définir l'intelligence (cf. paragraphe 1.1), qui est la propriété émergente du fonctionnement des neurones, ces derniers étant pourtant décrit par la biologie cellulaire. Dans un contexte plus informatique, nous parlerons dans ce document des champs de neurones dynamiques, dont le comportement global est difficile à qualifier, alors même que ce comportement est au cœur des processus de compétition sur lesquels repose nos recherches (cf. paragraphe 3.6.2). Dans notre cas, en conséquence, nous ne sommes même pas capables de construire le « langage L_2 » qui décrit les propriétés, ce qui pour l'instant exclut, malheureusement, nos recherches des tentatives actuelles de formalisation mathématiques de l'émergence basées sur la complexité des langages de description du système et de ses propriétés. En revanche, comme nous l'avons mentionné, l'approche de Kubík est en accord avec nos travaux, même si nous n'avons pas entrepris à ce jour d'études de cette nature.

En conclusion, nous soutenons par nos recherches que malgré les difficultés mentionnées, il est essentiel pour l'informatique de considérer les systèmes émergents comme l'un des moyen de produire des systèmes complexes, car cette approche, contrairement aux approches de conception classique, permet de programmer des systèmes sans que l'on ait à en spécifier les propriétés. Ainsi, dans le cas de la robotique autonome par exemple, spécifier les différents cas d'interaction entre le robot et son environnement est sujet à une explosion combinatoire rédhibitoire, et l'enjeu des systèmes émergents est de malgré tout pouvoir construire un contrôleur adapté à ces interactions, puisque sa conception ne repose pas sur l'analyse des cas d'utilisation. Dans le cadre de l'approche située de la cognition qui est la nôtre, cette conception de l'informatique a de plus l'avantage de potentiellement permettre aux systèmes d'accéder à l'intelligence, sans qu'il soit pour cela nécessaire de capturer ce concept dans quelque formalisme que ce soit, au risque inévitable de s'embourber dans l'inconsistance d'un tel projet.

Nous pensons que la maîtrise de la complexité en passant par le recours aux systèmes émergents est la seule voie consistante pour qu'un cerveau humain puisse concevoir une machine qui lui soit semblable.

Ceci est un autre enjeu de nos recherches, présenté au paragraphe suivant.

2.1.2 Machines intelligentes

Construire des machines intelligentes reste une des ambitions de l'Informatique, et nous avons, au chapitre précédent, fait la promotion de l'approche située pour aborder cette question. L'approche située toutefois, à elle seule, ne préjuge pas des mécanismes informatiques qui définissent le système en interaction avec son environnement, tant des architectures informatiques très diverses se sont revendiquées, à juste titre, de cette démarche scientifique. Ce que nous proposons par notre projet de recherche, c'est d'aborder la question de l'Intelligence Artificielle dans un cadre certes situé, mais en impliquant préférentiellement des mécanismes informatiques inspirés de la biologie du cortex. Le choix de ces mécanismes-là sera argumenté au paragraphe 2.2.2, mais il s'agit bien pour nous d'un a priori, dont nous pensons qu'il permettra de faire avancer la question de l'intelligence artificielle.

Nous proposons donc d'aborder l'Intelligence Artificielle par des systèmes incarnés, reposant sur une base connexionniste inspirée du cortex, qui mettent en avant l'émergence de propriétés présentes *implicitement* dans les mécanismes qui définissent le système. Dans ce contexte, les questions d'émergence du symbole et d'ancrage des raisonnements et des représentations évoquées au paragraphe 1.3.1 ne se poseront pas, puisque si symbole ou raisonnement il y a dans les

architectures que nous visons, il sera forcément ancré et situé. Ce n'est bien sûr pas pour autant que la difficulté de concevoir de tels systèmes est levée, et c'est tout l'intérêt pour l'informatique de poursuivre un engagement dans cette voie. Une approche de l'intelligence basée sur l'a priori de l'approche située et des systèmes émergents nous paraît être de nature à fonder le langage et le raisonnement sur un support de calcul analogue à celui évolué par la nature, et par là-même de construire des machines qui puissent nous aider à comprendre l'Homme. Cette ambition est bien moins défendable par les systèmes où la représentation est explicitée par le concepteur, comme le sont les ontologies et les systèmes à base de connaissance, puisque le fonctionnement est, par conception, radicalement différent de celui d'un cerveau humain.

2.1.3 L'exigence de calculs spatialement répartis

La fin de la loi de Moore, qui vise à extrapoler la croissance de la puissance des machines, est parfois annoncée, du fait que la technologie atteint les limites de la physique. Nous ne rentrerons pas ici dans ce débat, pour lequel des articles comme celui de Seth Lloyd [Lloyd, 2000] donnent le vertige quant à ce que l'on peut attendre théoriquement d'un ordinateur d'un volume d'un litre : 10^{51} opérations par seconde sur 10^{31} bits. Les arguments avancés dans ces débats sont de nature physique et technologique, et une distinction est à faire entre les limites théoriques et les verrous à lever pour effectivement augmenter la vitesse et la miniaturisation des processeurs actuels, comme ceux liés à l'agitation thermique [Kish, 2002]. Toutefois, l'une des voies envisageable d'évolution des processeurs est celle qui consiste non plus à accélérer encore la fréquence, mais à augmenter la surface des composants, comme c'est le cas des architectures *multi-core* qui commencent à proliférer aujourd'hui.

Il y a alors un enjeu pour l'informatique lié à cette évolution technologique qui est l'adéquation des calculs avec un support dont l'étalement spatial n'est plus négligeable au regard du temps de transmission des informations [DeHon, 2002]. Les échanges étant limités par la vitesse de la lumière, il y aura toujours une *latence* entre ce qui se passe à un endroit du composant et les effets provoqués à un autre endroit. Le système SCORE proposé par DeHon, par exemple, vise à expliciter le caractère spatial de tels calculateurs, basé sur le parallélisme des modules répartis dans l'espace et sur des connexions qui les hiérarchisent en accord avec leur position sur le silicium. Ces latences sont encore plus manifestes dans le cas d'un cluster, où les composants de calculs sont séparés de quelques dizaines de centimètres, et où les échanges sont gérés par des modules de routage. Si l'on admet donc une évolution de la puissance de calcul par un étalement croissant des composants sur l'espace, plutôt qu'une augmentation de leur fréquence, il convient pour profiter de cette puissance de concevoir des algorithmes qui puissent se répartir sur cet espace de calcul sans que les problèmes de latence ne constituent un goulot d'étranglement empêchant d'exploiter efficacement cet étalement spatial. Il s'agit d'une contrainte nouvelle en informatique, qui limite le nombre d'algorithmes qui pourront suivre efficacement une évolution technologique basée sur l'augmentation de la surface des composants de calcul.

Si l'on considère les neurones comme des unités de calcul, l'on constate qu'il s'agit d'une technologie extrêmement lente au regard des fréquences d'horloge de nos processeurs, puisque les neurones opèrent dans des domaines de fréquence de l'ordre de quelques dizaines de Hertz. La propagation de l'information se fait par des injections de courant au niveau des synapses, et des émissions de potentiel d'action, dont les vitesses de propagation sont de l'ordre de 20 mètres par seconde. Sur cette technologie, empreinte de phénomènes de latence considérables, la nature a su bâtir des systèmes nerveux extrêmement efficaces pour traiter les informations suffisantes à l'homéostasie et au comportement des êtres vivants ; la réponse de l'évolution est une répartition sur l'espace d'une énorme quantité de ces calculateurs « faibles ». Il ne s'agit alors plus de faire

circuler l'information dans quelques *pipelines* comme le font les microprocesseurs actuels, mais d'effectuer des calculs parallèles qui interagissent en permanence, de façon asynchrone, selon leur graphe de connexion. Ce graphe de connexion est de première importance, car il est soumis à la contrainte de pouvoir tenir, physiquement, dans l'enceinte limitée de la boîte crânienne, et ne peut donc être total.

Ces questions de gestion de la spatialité des calculs se posent également en informatique comme nous l'avons souligné, et notre projet de recherche propose une réponse par une approche connexionniste inspirée du cortex. Comme nous l'avons évoqué au paragraphe 2.1.1, le cortex peut être vu comme une répartition, sur une surface, de colonnes corticales, qui sont des calculateurs reliés entre eux par des liens de voisinage immédiats matérialisant les proximités spatiales, mais aussi des liens plus distants permettant des échanges avec des colonnes éloignées [Burnod, 1989]. Pour tenir dans la boîte crânienne en assurant une communication possible entre différentes parties du cortex, cette connectivité a une structure « *petit monde* » [Sporns, 2006]. Cette propriété semble d'ailleurs assez générale aux systèmes nerveux, puisqu'on la retrouve mise en avant dans des modèles de systèmes nerveux de « *simples* » vers [Berry and Temam, 2007] où elle permet une étude comparative de la dynamique du modèle neuronal avec des mesures biologiques. Les *réseaux petit-monde* ont été définis comme des réseaux réguliers, perturbés par une modification aléatoire de leurs connexions, créant ainsi des raccourcis [Watts and Strogatz, 1998]. Les auteurs soulignent que l'on trouve ces réseaux lorsqu'on analyse divers phénomènes, allant de la biologie que nous avons mentionnée aux réseaux sociologiques¹⁶, et il est clair que notre projet de recherche constitue une étude de plus parmi les innombrables illustrations de la richesse de ce concept.

Pour résumer, nous dirons qu'il existe une « *pression technologique* » d'une part, qui incite à étudier les systèmes qui exploitent la possibilité de certains calculs à être répartis spatialement, et que cette pression amène d'autre part les approches futures de la programmation à rencontrer les solutions déjà trouvées par la biologie, des systèmes nerveux en particulier. Dans notre cas d'étude des paradigmes de traitement d'inspiration corticale, présenté au paragraphe suivant, nous aurons à nous poser les questions d'implantation physique de nos calculs. Bernard Girau, avec qui nous collaborons, a par exemple répondu à cette question par l'utilisation de FPGA, appliqués à des modèles corticaux de traitement du mouvement visuel [Torres-Huitzil *et al.*, 2005], alors que pour notre part, nous avons choisi, du fait des collaborations que nous entretenons avec d'autres axes de recherche (cf. paragraphe 8.1.2), d'exploiter un paradigme de calcul instancié sur machines parallèles (cf. paragraphes 4.4.3 et 4.5). Dans ces deux cas, les enjeux de la spatialisation des calculs sont les mêmes, et les difficultés rencontrées sont de même nature.

2.2 Un projet de recherche informatique

L'objet de ce paragraphe est d'articuler notre projet de recherche, à la lumière de ce qui précède. Nous en présenterons les ambitions qui le motivent, ainsi que la « *réalisation* » à laquelle ces ambitions visent à aboutir.

16. Le terme « *petit-monde* » provient d'une étude de Milgramm, mettant en avant que le nombre moyen d'intermédiaires qui se connaissent deux à deux entre deux américains quelconques est 6.

2.2.1 Ambitions

Notre point de mire durant ces dernières années, et surtout pour celles qui s’annoncent, est d’être capable de donner corps aux idées les plus élégantes des sciences cognitives, que nous avons mentionnées au chapitre précédent. En particulier, les concepts de systèmes situés, de perception active, d’émergence de l’intelligence sans qu’il soit nécessaire de la spécifier, du langage et du raisonnement comme aptitudes initialement comportementales, sont des outils invoqués par les disciplines des sciences cognitives pour faire avancer la compréhension de « *l’esprit* » de l’Homme.

En informatique, nous pensons que comprendre quelque chose, c’est être capable de le produire, et c’est vers la production d’un programme au sein duquel les concepts que nous avons mentionnés puissent s’articuler que nous nous dirigeons.

Pour atteindre cet objectif, nous avons besoin d’un fil conducteur, dont on sait qu’il permet d’aboutir. Ce fil est le cerveau et ce que l’on en connaît, ce qui nous amène à étudier les approches neuro-mimétiques en informatique. Plus précisément, nous nous intéressons au cortex, ce que nous argumenterons au paragraphe suivant. Notre projet de recherche vise donc à concevoir un *calcul cortical* situé, instanciant sur silicium, et de façon opérationnelle, les idées des sciences cognitives dont nous avons fait la promotion au chapitre précédent. Dans ce contexte, la robotique est de première importance, puisque nous parlons de systèmes situés, ce qui, sans être impossible, est difficilement concevable sans *l’incarnation* d’un agent dans son environnement. Le rôle que nous attribuons à la robotique dans ce projet est à distinguer des approches automatiques, où le contrôle d’un système est central, et où les équations de ce système sont explicitées. Pour notre projet, les robots sont un support d’incarnation, et le fait qu’une tâche puisse aussi être résolue par des méthodes d’automatique, de façon aujourd’hui plus efficace en terme de performances (répétabilité, fiabilité, etc.) n’est pas une objection à notre projet. En effet, les approches automatiques actuelles, aussi pertinentes qu’elles soient, ne répondent pas du tout aux questions que nous avons soulevées.

Notre démarche s’efforce donc de proposer un modèle inspiré du cortex, qui se voit par conséquent confronté à des problèmes de complexité, en terme de nombre d’unités de calcul, de connectivité, de mécanismes d’évaluation. Il nous faut proposer des solutions, informatiques, pour maîtriser cette complexité. Nous nous ramenons-là à des problèmes plus classiques de génie logiciel, et nous proposons d’instancier notre projet sous forme de bibliothèques logicielles, autorisant la définition (cf. paragraphes 4.4.3 et 4.5) puis le portage de nos modèles sur machines parallèles (cf. paragraphe 8.1.2), comme nous l’avons mentionné précédemment.

Enfin, nous rappellerons ici les rapports qu’entretient ce projet avec la réalité biologique du cortex. Nous revendiquons une inspiration du cortex, sans nier qu’elle est sujette à notre interprétation. Cette revendication est à considérer comme un hommage aux travaux des scientifiques qui ont étudié le cortex, et n’est pas l’affirmation de la plausibilité de notre travail par rapport au cortex. Ce dernier point nous apparaît être du ressort d’un biologiste bien plus que d’un informaticien, et aussi, dans les progrès de nos recherches, nous efforcerons-nous de rendre ce travail accessible aux biologistes, afin qu’ils puissent porter ce jugement de plausibilité sur nos travaux, et peut-être y trouver matière à faire progresser la compréhension de leur propre discipline.

2.2.2 Les ressources de calcul extensives comme objectif

Notre choix de nous limiter au cortex comme source d'inspiration de nos recherches est fondé sur deux arguments. Le premier est qu'il ne nous paraît pas envisageable de se consacrer à l'ensemble du cerveau, même si cela est tentant lorsqu'on aborde des tâches robotiques, qui requièrent de la mémoire spatiale, de la mémoire de travail, de la mémoire épisodique, ... prises en charge en biologie par l'ensemble du système nerveux. Nous sommes conscient qu'il est artificiel d'isoler une structure du reste du système nerveux, mais il s'agit là de notre part d'un choix méthodologique. Le deuxième argument est celui qui dirige notre choix vers le cortex alors que d'autres structures sont tout aussi complexes et intéressantes pour la compréhension des capacités de traitement des structures neuronales. Cet argument se fonde sur les propriétés d'auto-organisation prêtées au cortex, sur lesquelles nous allons nous attarder.

Nous argumenterons de façon plus étayée au chapitre 4 que le cortex est une structure qui rassemble des unités de calcul relativement homogènes, les colonnes corticales, sur une surface chiffonnée dans notre boîte crânienne. Nous voyons le cortex comme « *du calcul au m²* », pouvant être *recruté* au gré des besoins du comportement. Ainsi, s'il faut traiter une information visuelle, qui anatomiquement se projette sur le cortex occipital, cette partie de la surface corticale se dédie au traitement visuel, alors que le même substrat, chez les aveugles, sert à l'intégration de palpations tactiles. Le cortex apparaît comme une surface capable d'une allocation spontanée de ressources quand cela est nécessaire, ces ressources ayant avant de se spécialiser la potentialité de traiter des problèmes de modalité diverses.

Nous visons par nos recherches la définition d'une unité de calcul élémentaire qui, une fois dupliquée massivement sur une surface, puisse constituer un module capable de se spécialiser à réaliser une tâche. L'idée de construire des systèmes de cette façon se retrouve dans les travaux sur l'*amorphous computing* [Abelson *et al.*, 2000], qui proposent de constituer des peintures intelligentes qui contiennent des myriades de cellules pouvant s'organiser pour prendre en charge une fonction. Ces idées sont inspirées du développement cellulaires de l'embryon, et les auteurs proposent une bibliothèque pour les mettre en œuvre [Coore, 2005].

Notre démarche est analogue, si ce n'est que c'est le paradigme de l'auto-organisation corticale qui nous guide, plutôt que celui de la différenciation cellulaire. Mais il est clair que la biologie regorge d'exemples de ce type, et que sa « *technologie* » exploite la construction des systèmes par auto-organisation de constituants ayant les propriétés adéquates à l'émergence de fonctions élaborées.

« *We turn to biology not just as a metaphor, but as an actual implementation technology for amorphous systems by means of « cellular computing, » which constructs logic circuits within living cells.* » .

[Abelson *et al.*, 2000]

Nous proposons de promouvoir la notion de *ressources de calcul extensives*, à propos desquelles nous participons à un groupe de travail avec l'équipe Cortex du Loria¹⁷. L'idée est de pouvoir proposer à un utilisateur *de la* ressource de calcul au mètre carré. Cette ressource s'adapterait à son problème, qui serait plutôt la prise en charge par une machine de compétences sensorimotrices. Si l'utilisateur estime les performances insuffisantes, ou s'il souhaite ajouter une autre modalité à son système, comme des capteurs ou des effecteurs supplémentaires, il lui « *suffirait* » de rajouter de la surface de calcul, et le système se réorganiserait pour prendre en charge la nouvelle coordination de l'ensemble. Ce scénario est imagé par la figure 2.1.

17. Principalement Frédéric Alexandre, Yann Boniface, Bernard Giraud, Nicolas Rougier avec qui nous tentons de mûrir cette idée. Nous avons par exemple soumis un projet ANR en 2005, qui n'a pas été retenu.

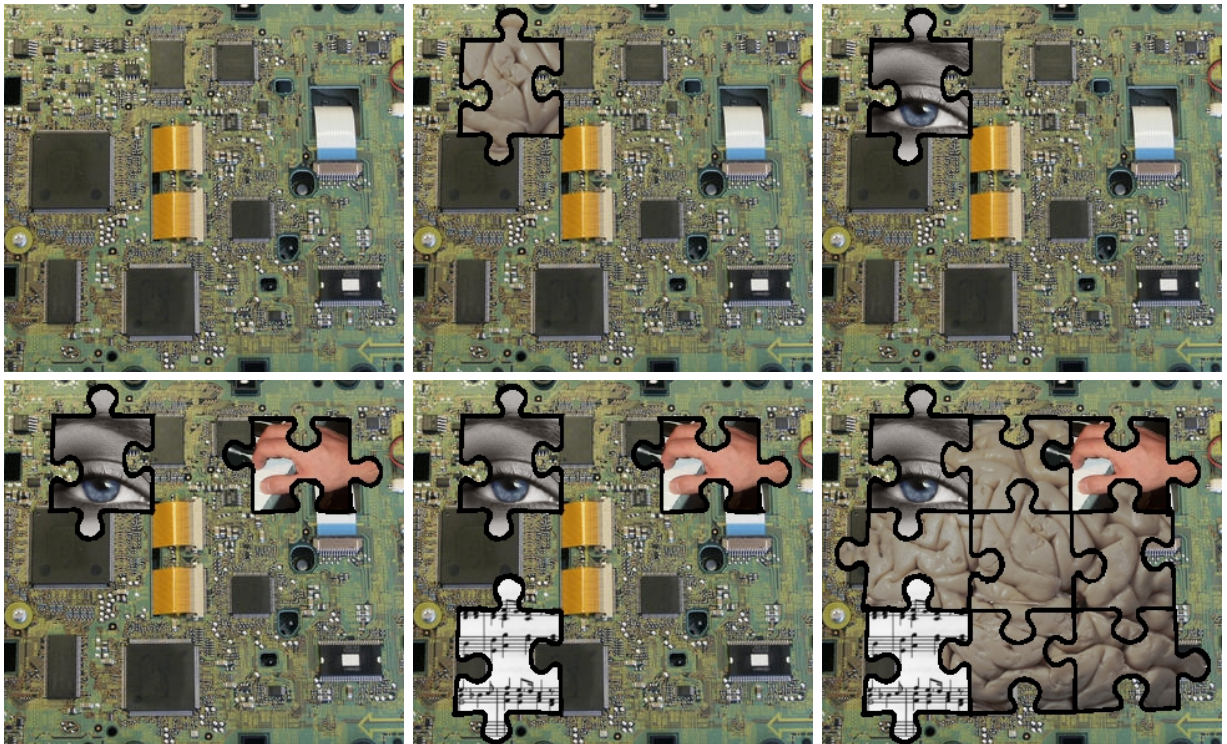


FIGURE 2.1 – Scénario schématique d’allocation de ressources extensives. Dans le sens de lecture : Définition d’un composant physique. Allocation d’une ressource de calcul indifférenciée. Cette ressource peut s’interfacer avec une entrée, disons visuelle. Une autre ressource, du même type peut s’interfacer avec des effecteurs (main), où un capteur d’une autre modalité (auditive). Enfin, d’autres ressources, de même nature que les précédentes, peuvent être mobilisées pour prendre en charge une gestion multimodale de l’ensemble des informations.

Se posent alors avec la plus vive acuité les questions relatives à la standardisation des interfaces entre ces modules et à la spécification de leur fonctionnalité. Comment les connecter entre eux ? Comment les interfacer avec des capteurs ? Comment les soumettre à une tâche pour qu'ils s'auto-organisent ? Que se passe-t-il au niveau des modules existants si l'on rajoute de la ressource de calcul ? Toutes ces questions sont bien sûr encore ouvertes, et les réflexions que nous menons avec nos collaborateurs sur ce point sont en cours. Toutefois, dans ce document, nous présentons le modèle `bijama` développé au cours de la thèse d'Olivier Ménard [[Ménard, 2006](#)], qui nous apparaît esquisser des réponses possibles à ces questions.

Chapitre 3

Systèmes Dynamiques

Les systèmes dynamiques, sur lesquels nous allons nous attarder dans ce chapitre, permettent de modéliser des phénomènes naturels que nous jugeons complexes. Le lecteur intéressé par la capacité de ces systèmes à décrire des phénomènes complexes pourra se reporter à l'excellent ouvrage de vulgarisation scientifique de James Gleick [Gleick, 1991]. Les systèmes dynamiques ont également, dans le cas des systèmes chaotiques, montré que le caractère déterministe d'un processus ne suffit pas à le rendre prédictible, ce qui n'est pas si intuitif de prime abord. Enfin, et c'est là que nos recherches sont concernées par ces systèmes, il est très facile, lors de la conception d'un système, de tomber sur un système dynamique au comportement « *complexe* », du moment qu'on ne s'impose pas de rester dans les ornières des systèmes linéaires. La quantité de résultats mathématiques disponibles pour les systèmes linéaires est en effet un bon argument pour rester dans ce cadre, et l'approximation d'une réalité par un système linéaire est souvent pertinente. Toutefois, dans le cadre de nos recherches, nous mettons en œuvre des unités de calcul qui sont d'emblée non linéaires, dont les comportements sont couplés, ce qui nous conduit à concevoir des systèmes dynamiques, et à nous confronter par conséquent au caractère particulier des outils qui les décrivent.

3.1 Formalisation

Nous donnons ici quelques définitions des systèmes dynamiques, à partir de celles fournies dans la littérature [Beer, 2000; Meiss, 2007]. Les références aux travaux fondateurs de Henri Poincaré sont rappelées dans [Meiss, 2007].

3.1.1 Définitions

Un *système dynamique* est la donnée d'un espace d'états S , appelé aussi espace des phases, d'un ensemble des temps T , et d'une fonction Φ qui caractérise l'évolution de l'état du système au cours du temps. Nous distinguerons deux classes de systèmes dynamiques, ceux qui sont à temps discret et ceux à temps continu.

Les systèmes à temps discret, pour lesquels $T = \{t_0, t_1, \dots\}$, sont définis par $\phi : S \rightarrow S$, qui permet de construire une suite d'états $x_i \in S$, $i \in T$, par la récurrence $x_{i+1} = \phi(x_i)$. Pour un x_0 donné, la suite ainsi construite, à partir de x_0 , est appelée l'*orbite* de x_0 .

Pour les systèmes à temps continu, l'évolution est définie par $\Phi : S \times T \rightarrow S$, dont la signification est que $x = \Phi(x_0, t)$ est l'état dans lequel se retrouve, au temps t un système qui était en x_0 au temps $t = 0$. Dans ce cas, en supposant que Φ ait les propriétés requises de

différentiabilité, on étudie le *champ de vecteurs* X défini par l'équation 3.1.

$$\frac{dx}{dt} = X(x(t)) \quad (3.1)$$

Un système soumis à une perturbation « *externe* » $e(t)$ est de même décrit par l'équation 3.2.

$$\frac{dx}{dt} = X(x(t), e(t)) \quad (3.2)$$

3.1.2 Attracteurs et points fixes

Vers quoi évolue le système ?

Selon Φ , les orbites du système peuvent avoir la propriété d'atteindre un point de repos, c'est à dire pour lequel x reste constant. L'état x est appelé alors *point fixe* du système dynamique, et il vérifie, selon nos notations dans les deux cas des systèmes à temps discret et continu, $\phi(x) = x$ ou $X(x) = 0$.

Si une orbite n'atteint pas un point fixe, et si l'on suppose que les états possibles que peut prendre le système sont inclus dans un ensemble borné, l'orbite peut avoir des comportements « *bizarres* ». Tout d'abord, si S est un espace fini, une telle orbite finit nécessairement par revisiter, au bout d'un temps t , un état x_a visité antérieurement. Le système étant déterministe, on peut en conclure que l'orbite est cyclique, le cycle étant celui décrit entre deux passages par x_a . Dans le cas d'un espace d'état continu, on peut également avoir des orbites cycliques, mais, en gardant S borné, on ne peut plus conclure à l'existence de cycles. En effet, les orbites peuvent sillonner S à l'infini sans jamais revisiter un état. Parmi ces systèmes, ont été décrits dans la littérature des systèmes qui présentent des orbites de ce type qui sont « *bien localisées* », c'est-à-dire qui ne sillonnent pas tout l'espace S . Intuitivement, et nous n'irons pas plus loin dans ce document, ces orbites sont des courbes infiniment longues, mais malgré tout confinées dans l'espace, et leur structure a « *en conséquence* » des propriétés fractales. L'ensemble des orbites de ce type localisées au même endroit est ce qu'on appelle un *attracteur étrange*.

L'évolution du système dynamique finit par atteindre un point fixe, rejoindre un cycle, où être capturée dans un attracteur étrange. On parle de *bassin d'attraction* d'un point fixe pour désigner l'ensemble des états de S qui conduisent le système vers ce point fixe. On définit de même des bassins d'attraction pour les cycles et les attracteurs étranges.

Exposants de Lyapunov et Chaos déterministe

Prenons le cas célèbre de l'*attracteur de Lorenz* [Lorenz, 1963], dont la genèse est racontée dans [Gleick, 1991], afin d'illustrer le concept de *chaos déterministe*. Cet attracteur est l'attracteur étrange du système dynamique ci-dessous, dont l'espace d'état est \mathbb{R}^3 , défini par (équation impliquant le champ de vecteur) :

$$\begin{aligned} \frac{dx}{dt} &= \sigma(y - x) \\ \frac{dy}{dt} &= x(\rho - z) - y \\ \frac{dz}{dt} &= xy - \beta z \end{aligned} \quad (3.3)$$

Pour $\sigma = 10$, $\beta = 8/3$ et $\rho = 28$, on constate que les orbites se font capturer par l'attracteur illustré sur la figure 3.1. Ce qui est remarquable est la structure fractale de l'attracteur. En fait, les trajectoires qui restent sur le même œil et celles qui s'échappent vers l'autre sont « *infinitement entremêlées* », et il faudrait alors disposer d'une précision infinie sur l'état de départ du système pour prédire son avenir... alors qu'il s'agit bien d'un système déterministe! C'est en ce sens que le système est défini comme chaotique.

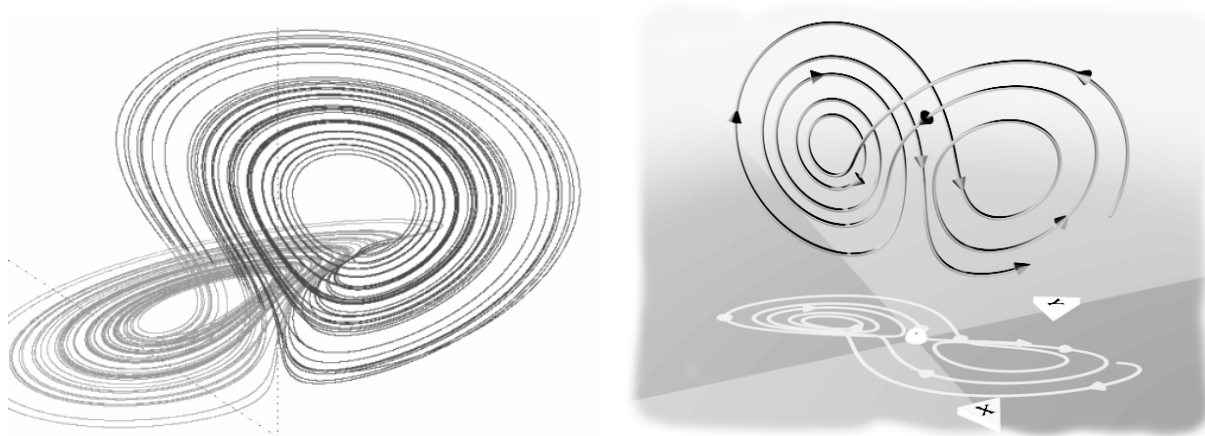


FIGURE 3.1 – Attracteur étrange de Lorenz. Dans les deux figures, le dessin de cet attracteur résulte de la trace d'une orbite du système dynamique.

Un système chaotique est alors un système, pour lequel deux points initialement proches peuvent finir par s'éloigner. Cette caractéristique est définie mathématiquement par les *exposants de Lyapunov* du système. Dit brièvement, on considère à un temps $t = 0$ deux points $x = x(0)$ et $y = y(0)$ de S , que l'on fait évoluer dans le temps. On dispose ainsi de la grandeur $\Delta x(t) = |x(t) - y(t)|$, et on appellera Δx l'écart initial $|x - y|$. L'exposant de Lyapunov λ en x est alors défini par l'équation 3.4. Dit autrement, l'écart $\Delta x(t)$ entre deux orbites infiniment proches évolue comme $\Delta x.e^{\lambda t}$.

$$\lambda = \lim_{\substack{t \rightarrow \infty \\ \Delta x \rightarrow 0}} \frac{1}{t} \ln \frac{\Delta x(t)}{\Delta x} \quad (3.4)$$

En fait, une orbite partant d'un point dont l'exposant de Lyapunov est strictement négatif est dissipative, le système « *s'arrête* », alors que si ce coefficient est nul, le système est conservatif, l'orbite peut être un cycle par exemple, provoquant une oscillation régulière indéfiniment. Enfin, si l'exposant est positif, l'on a affaire à une instabilité du système, qui est chaotique en ce point. Le plus gros exposant de Lyapunov des points de S (dit MLE¹⁸) permet de donner une caractéristique globale du système.

En pratique, il existe des méthodes numériques pour estimer les coefficients de Lyapunov d'un système, ou simplement le MLE, mais nous ne rentrerons pas dans ces développements ici.

Bifurcations

On s'intéresse souvent, lorsque l'on met en œuvre un système dynamique, à ses *bifurcations*. Les bifurcations sont un changement de la dynamique d'un système, consécutives à un change-

18. Maximum Lyapunov Exponent.

ment des paramètres de ce système. Si l'on revient au cas de l'attracteur de Lorenz (cf. équation 3.3), jouer sur le paramètre ρ conditionne l'apparition ou non de l'attracteur. Un point de bifurcation est alors un point de *l'espace des paramètres*¹⁹ autour duquel le système change le nombre de ses points fixes, de ses attracteurs. La stabilité des points fixes, la nature chaotique ou non des attracteurs, peuvent changer également.

3.1.3 Stabilité

Nous avons déjà évoqué la stabilité des systèmes dynamiques en mentionnant la définition des exposants de Lyapunov. En pratique, c'est-à-dire pour nous dans le cadre des modélisations de systèmes neuronaux comme celles rassemblées dans [Gerstner and Kistler, 2002], on s'intéresse également à la stabilité des systèmes en leurs points fixes. En ces points, le système est stable si une perturbation autour du point fixe conduit à ramener le système en ce point fixe. On mesure cette stabilité en linéarisant l'expression du champ de vecteur autour du point fixe. En fait, si x_0 est un point fixe, la stabilité est liée aux valeurs propres de la jacobienne du champ X de l'équation 3.1. On a ainsi des points que l'on peut démontrer comme stables (valeurs propres aux parties réelles négatives), instables (valeurs propres positives), ou aux propriétés plus complexes (point-selle, etc.).

De façon plus générale, on peut, pour étudier la stabilité d'un système, essayer d'en exprimer une *fonction de Lyapunov*. Il s'agit d'une fonction, disons $V : S \rightarrow \mathbb{R}$ qui à chaque état x associe une « énergie » $V(x)$. Pour démontrer la stabilité du système en un point fixe x_0 , il *suffit* de choisir V (et il n'y a pas de méthode générale pour guider ce choix) telle que pour tout point x dans un voisinage de x_0 ,

- on ait $V(x) - V(x_0) \leq 0$, avec l'égalité atteinte pour $x = x_0$ seulement,
- et la variation d'énergie $\frac{dV}{dt}$ du système qui soit négative.

Dit autrement, on peut montrer la stabilité d'un système en ses points fixes en exhibant une fonction de Lyapunov pour laquelle les points fixes se situent en des minima locaux, comme schématisé sur la figure 3.2. Si l'on arrive à assurer, par la forme de V , que $\forall x \in S, \frac{dV}{dt} \leq 0$ (on étend la condition du voisinage des points fixes à tout S), alors on peut montrer que l'énergie du système ne cesse de diminuer, jusqu'à atteindre un minimum sur un point fixe. Cela donne une preuve de convergence du système dynamique. Pour vérifier la négativité de la dérivée temporelle de V , il suffit de poser l'équation 3.5.

$$\frac{dV}{dt} = \frac{\partial V}{\partial x(t)} \times \frac{dx}{dt} = \text{grad}_x^{(V)}(x(t)) \cdot X(x) \quad (3.5)$$

Avec les fonctions de Lyapunov, les bassins d'attractions des points fixes se traduisent en bassins dans le paysage énergétique, et la dynamique du système revient à descendre, dans ce paysage, la pente du bassin courant jusqu'à se retrouver en son minimum, le point fixe.

3.2 Modélisation par les systèmes dynamiques : encore un problème de robinet

Nous avons mentionné que les systèmes chaotiques sont des systèmes fondamentalement imprévisibles, alors même qu'ils sont déterministes. Dans la mesure où notre travail de recherche implique des étapes de modélisation, nous nous intéressons dans ce paragraphe à la particularité de la modélisation de systèmes dynamiques. En effet, si le modèle est un système dynamique

19. À ne pas confondre avec l'espace de phases.

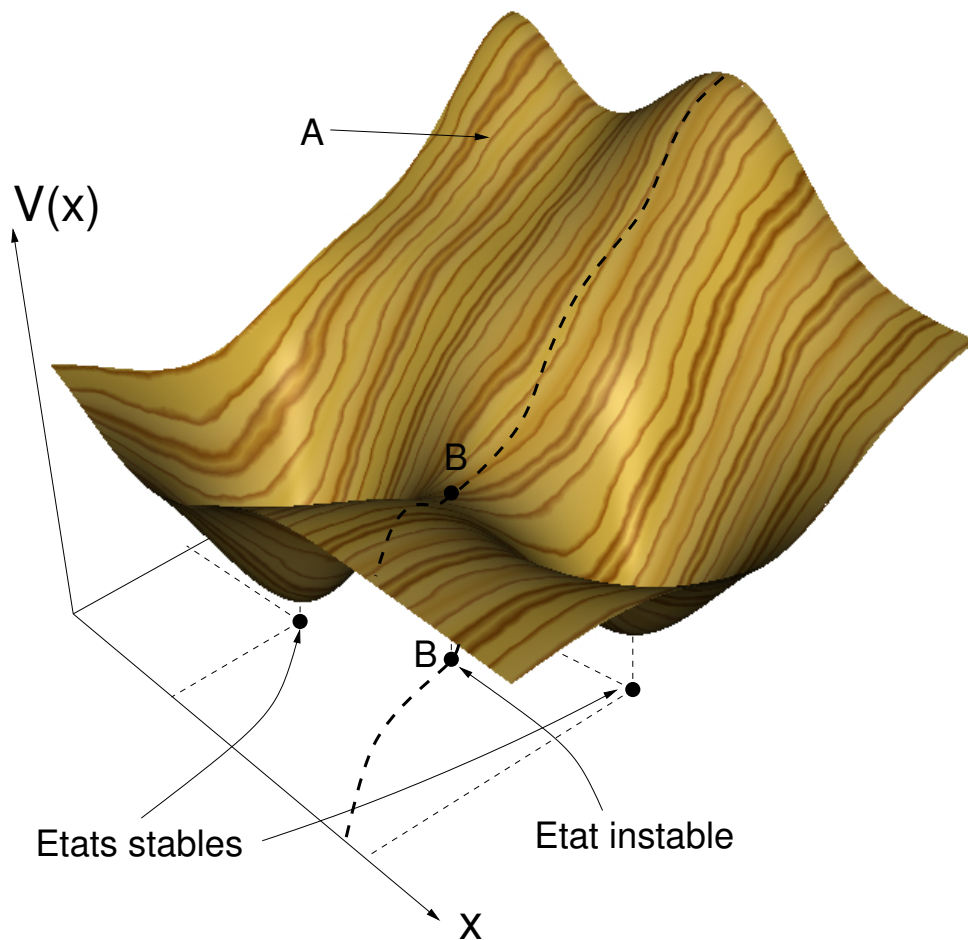


FIGURE 3.2 – Vue schématique des valeurs $V(x)$ d'une fonction de Lyapunov d'un système dynamique. L'espace des états x , i.e. espace des phases, est le plan horizontal. Si le système est au point A , il sera attiré par l'attracteur le plus à gauche de la figure. Le point B est un point selle de V ; il s'agit d'un point d'équilibre instable. La frontière qui sépare les bassins d'attractions des deux points fixes, i.e. états stables, est représentée en pointillés.

chaotique, il y a peu de chance qu'il puisse reproduire des données... Il est alors clair qu'il faut un autre argument que celui de la comparaison directe d'une simulation du modèle avec la réalité pour juger de la qualité du modèle.

Cet argument tient à la comparaison des attracteurs du modèle et du système réel. Un bon modèle est alors, dans ce sens, un modèle dont l'attracteur « *ressemble* » à l'attracteur mesuré sur le phénomène réel. Cet argument nous paraît extrêmement puissant, et évite de tomber dans le piège de comparer les orbites d'un modèle avec la suite d'états observés pour le système réel, ces deux objets étant de toute façon très différents dans le cadre d'un système chaotique.

À l'instar de James Gleick [Gleick, 1991], revenons à un exemple cher aux mathématicques de l'écolier : le robinet qui fuit. Ce système ne se contente pas d'exiger qu'une règle de trois donne le temps nécessaire pour remplir, à grand renfort de gouttelettes, une baignoire d'un volume donné, car il a également un comportement chaotique, comme suggéré par Otto Rössler et montré par Robert Shaw. En effet, les gouttes qui se forment à l'extrémité sont de masse régulièrement croissante, puisqu'elles se remplissent d'eau, jusqu'à céder. Toutefois, cette goutte en formation oscille, et si la masse critique de rupture est atteinte alors que la goutte remonte, la rupture aura lieu à la descente suivante, avec une goutte plus massive.

Afin d'exhiber le caractère chaotique de ce système, les auteurs ont choisi de construire un diagramme, appelé *return map*, comme suit. On enregistre d'abord une séquence d'observations des états successifs du système, ici la séquence des écarts dt_i entre les instants t_i et t_{i+1} où les gouttes i et $i + 1$ se décrochent. Le diagramme se construit en traçant les points (dt_i, dt_{i+1}) . Si l'on réalise l'expérience physique du robinet [Néda *et al.*, 1996], on voit, pour certains paramètres, apparaître sur le diagramme le dessin d'un attracteur étrange.

Ce problème est très difficile à modéliser, tant les phénomènes de tension superficielle permettant de déterminer les points de rupture de la goutte sont délicats. On peut toutefois proposer un modèle grossier, utilisant une masse croissante suspendue à un ressort, et l'on simule le décrochement de la goutte par une réduction brutale de cette masse. Il a été vérifié sur des simulations qu'un bruit dans les paramètres du modèle conduit à un attracteur qui est une version « *floue* » de l'attracteur idéal [Iarraza-Lomelí *et al.*, 1999], comme illustré sur la figure 3.3. En fait, on peut utiliser un modèle aussi réducteur que celui d'une masse au bout d'un ressort pour simuler le système réel, et l'ajustement des paramètres peut se faire en comparant les attracteurs obtenus par le modèle avec la réalité [D'Innocenzo *et al.*, 2006].

Il est ainsi possible de contrôler la période d'écoulement des gouttes [Iarraza-Lomelí *et al.*, 1999], ce qui est utilisé pour les imprimantes à jet d'encre. En ce qui nous concerne, nous retiendrons de cet exemple qu'un modèle simpliste peut exhiber des dynamiques chaotiques dont le comparaison des attracteurs avec ceux du modèle réel est un argument de validité.

3.3 Automates cellulaires

Les *automates cellulaires* sont un cas particulier de systèmes dynamiques qui est particulièrement proche de la nature des systèmes dynamiques que nous étudions dans le cadre de nos recherches. Un automate cellulaire est une machine composée de cellules, dont l'état se résume, à chaque instant discret, à l'ensemble des états de ses cellules. L'état d'une cellule change à certains pas de temps, ce qui correspond à une *mise à jour*, qui est dans ces systèmes dépendante de l'état des autres cellules de l'automate.

Ce qui nous intéresse plus particulièrement dans le cadre de nos travaux est ce que nous appellerons le *champ récepteur*²⁰ d'une cellule c d'un automate. Selon notre définition, il s'agit

20. La notion de champ récepteur en biologie est un peu différente, l'utilisation de ce terme ici n'est qu'une

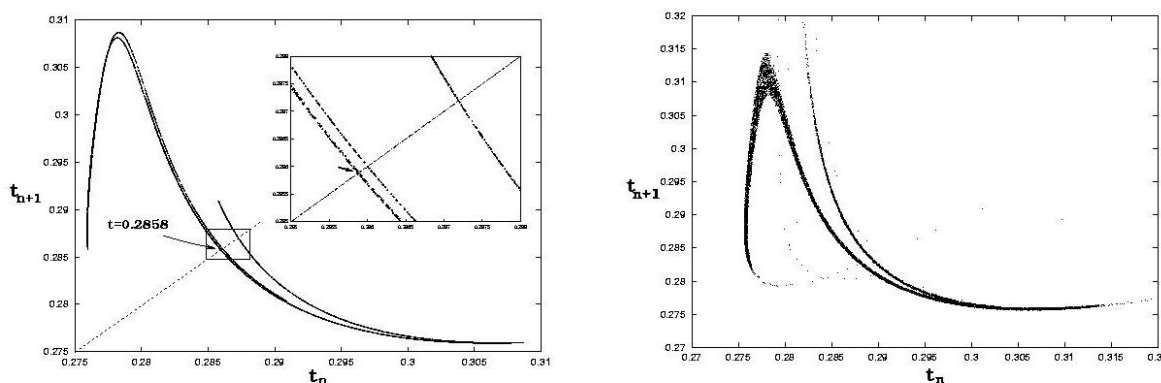


FIGURE 3.3 – Diagramme *return map* d'un modèle masse-sur-ressort du robinet qui fuit (à gauche), d'après [Iarraza-Lomeli *et al.*, 1999]. On constate que le bruitage des paramètres du modèle rend l'attracteur flou (à droite), ce qui laisse penser que la comparaison d'un attracteur mesuré sur le système réel avec l'attracteur idéalisé du modèle est pertinente pour identifier des paramètres du système réel par exemple.

de l'ensemble des cellules de l'automate dont l'état est lu par c pour effectuer son changement d'état, sa mise à jour. Le champ récepteur d'une unité, qui est une propriété de l'architecture de l'automate, reste le même au cours du temps. On peut représenter alors l'*architecture* d'un automate cellulaire par un graphe orienté, dont les nœuds sont les cellules, et dont les arêtes relient une unité à chacune de celles qui appartiennent à son champ récepteur. Nous dirons que l'existence de l'arête $A \leftarrow B$ signifie que A lit l'état de B pour faire sa mise à jour. C'est la nature des règles de mise à jour ainsi que l'architecture de l'automate qui permettent d'envisager par ce type de systèmes dynamiques la mise en œuvre des ambitions exprimées au paragraphe 2.2.

Le cas le plus élémentaire est celui des automates dont les cellules sont booléennes. Le célèbre *Jeu de la vie* de Conway (1970), est un réseau de topologie torique, où chaque cellule, booléenne, est connectée à ses 8 voisins dans une grille. Muni d'une règle de mise à jour appliquée en parallèle à chaque cellule, l'automate ainsi obtenu a la puissance de calcul nécessaire pour implémenter une machine de Turing [Rendell, 2002]. Tout se passe en étalant, sur la surface des cellules, un état, dont l'évolution par application locale de règles simples peut correspondre à un calcul très complexe.

3.3.1 Là où le bât blesse

Même en nous restreignant au cas particulier de cellules à deux états, se posent d'ors et déjà deux difficultés majeures. La première est qu'il n'est pas évident de trouver une règle de mise à jour qui permette d'exhiber des propriétés de calcul intéressantes [Gutowitz and Langton, 1988], comme nous l'avons déjà mentionné page 29, et il n'y a pas de méthodologie, à notre connaissance, pour guider mathématiquement cette conception. La deuxième difficulté réside dans le fait que tout état de départ ne permet pas d'exhiber des calculs intéressants. C'est un état très particulier, conçu « *à la main* », qui permet d'implémenter, comme l'a fait Rendell, une machine de Turing sur la base du jeu de la vie.

Devant l'absence de méthodologie formelle, nous avons choisi pour méthodologie de nous inspirer de ce que nous pouvons apprendre de l'organisation du tissu cortical, ce qui suppose analogie.

de considérer des automates plus complexes que ceux que nous avons présentés jusqu'ici. Cette extension de complexité se fait suivant deux directions : le déroulement des mises à jour d'une part, et la nature des cellules d'autre part.

3.3.2 Mises à jour asynchrones

Nous avons implicitement considéré jusqu'ici, du fait de notre exemple du jeu de la vie, des *mises à jour synchrones*. Il s'agit, à chaque pas de temps, de mettre à jour toutes les cellules en parallèle. Chaque cellule, pour calculer son état au temps $t + 1$, dispose de l'état au temps t des cellules de son champ récepteur. Sur machine séquentielle, on peut simuler ce parallélisme en sauvegardant l'état au temps t dans un tampon auxiliaire, utilisé pour les lectures lors de la mise à jour.

Cette mise à jour parallèle possède toutefois un inconvénient majeur, que nous allons illustrer par l'exemple suivant. Supposons un automate dont chaque cellule peut prendre l'état 0 ou 1, et dont chaque cellule possède l'ensemble de toutes les cellules comme champ récepteur (connectivité totale). Nous souhaitons définir la même règle d'activation pour toutes les cellules, et souhaitons que l'automate assure qu'exactly n cellules soient dans l'état 1. Une règle simple de mise à jour est, pour chaque cellule, de faire la somme des valeurs des cellules de son champ récepteur (toutes les cellules donc), comptant ainsi le nombre de cellules ayant la valeur 1, et de passer à 1 si cette somme est plus petite que n , de passer à 0 si cette somme est plus grande que n , et de garder son état courant si cette somme vaut exactement n . Il est clair que tous les états ayant exactement n cellules à 1 sont des points fixes de l'automate. En revanche, dans les autres cas, on aboutit à une oscillation. En effet, si le nombre de cellules à 1 est plus grand que n , chaque cellule, appliquant la mise à jour énoncée précédemment, va passer à 0. À la mise à jour suivante, vu que toutes les cellules de l'automate sont à 0, leur somme est nulle, et la mise à jour consistera à passer toutes les cellules à 1, puis toutes à 0 au pas de temps suivant, etc. Les points fixes sont donc particulièrement instables, car la moindre perturbation conduit à une orbite de l'automate cellulaire oscillant entre les deux états homogènes possibles.

Considérons à présent la règle de mise à jour inspirée de celle utilisée dans les réseaux de Hopfield [Hertz *et al.*, 1991], que nous qualifierons de *mises à jour asynchrones*. Idéalement, il s'agit de choisir à chaque pas de temps une cellule au hasard, et de lui appliquer une mise à jour. Nous préférons assurer que le temps s'écoule à la même vitesse pour toutes les cellules, et la règle de mise à jour asynchrone que nous définirons consiste, à chaque cycle, à définir une permutation aléatoire des indices des cellules, puis à appliquer une mise à jour par cellule, dans l'ordre défini par la permutation. Une fois un cycle terminé, chaque cellule a été mise à jour une fois et un autre cycle est exécuté.

Dans le cas de notre automate, la mise à jour asynchrone permet d'atteindre un point fixe proche de la configuration initiale. Les premières cellules activées lors d'un cycle vont passer à 1 ou 0 selon qu'il faille ajouter ou retirer des 1 dans l'automate pour atteindre la valeur n , mais une fois cette valeur atteinte, toutes les mises à jour suivantes conservent l'état des cellules.

On constate sur cet exemple que le passage à l'asynchronisme a permis d'éliminer des cycles, et a étendu le nombre d'états à partir desquels le système se fait attirer vers l'un de ses points fixes. Cet effet de l'asynchronisme, même lorsque cet asynchronisme est partiel²¹, est assez général aux systèmes dynamiques discrets, et est progressif avec la baisse de parallélisme au sein d'un cycle. Nous renvoyons le lecteur aux concepts de *graphes d'itérations* et d'*opérateurs de Gauss-Seidel* présentés dans [Robert, 1995] pour des justifications plus rigoureuses de ce point.

21. Asynchronisme partiel signifie qu'un cycle est constitué d'une succession d'activations parallèles d'ensembles de cellules, ces ensembles formant une partition de l'automate.

L'exemple que nous avons pris est un cas simple de décision collective au sein de la population des cellules, à savoir que les cellules doivent « *s'arranger* » collectivement pour qu'un critère de population, le nombre total de cellules à 1, soit rempli. Cette notion de décision collective est au cœur des mécanismes sur lesquels sont basées nos recherches, et c'est pourquoi nous avons recours à des mises à jour asynchrones. Pour concevoir des systèmes constitués de nombreuses unités, il convient de noter que la mise à jour asynchrone est difficile à paralléliser sur un cluster, ce qui motive nos collaborations avec Stéphane Vialle de Supélec et Jens Gustedt de l'équipe Algorille du Loria (cf. paragraphe 8.1.2).

3.3.3 Automates à états continus

La notion d'automate cellulaire, encore aujourd'hui, renvoie en général à des automates dont les cellules ont des activations booléennes. Il existe toutefois des extensions aux cas d'activations continues, que nous présentons ici. Néanmoins, si l'automate à activations continues est exécuté sur un ordinateur, qui est par essence une machine discrète et binaire, il est clair que cet automate est, in fine, un automate à activations booléennes, comme le fait remarquer François Robert [Robert, 1995]. La remarque, toutefois, ne tient plus lorsque les cellules de l'automate sont physiquement instanciées sous formes de circuits électroniques analogiques.

C'est du domaine de l'électronique que nous viennent les premières formulations de ce qu'on appelle les *Cellular Neural Networks* (CNN), puisqu'il s'est agit d'implanter sur composants VLSI des circuits génériques, permettant la résolution d'équations aux dérivées partielles [Chua and Yang, 1988; Sargeni and Bonaiuto, 2005]. Ces études se basent sur la méthodologie suivante. Tout d'abord, les dérivées spatiales de l'équation différentielle sont remplacées par des différences finies, créant ainsi des relations entre les cellules de l'automate qui correspondent à des points proches du maillage de l'espace. Restent alors uniquement les dérivées en temps qui, pour chaque cellule de l'automate, déterminent son évolution en fonction de ses voisines, ce qui décrit l'évolution d'un système dynamique comme celle formalisée par l'équation 3.1 page 38 [Roska *et al.*, 1995; Kozek *et al.*, 1995].

La motivation pour l'étude des CNN est clairement la résolution d'équations pour la physique, nous reviendrons sur ce point au paragraphe 6.1. Toutefois, et c'est en ce sens que les CNN sont à rapprocher de nos recherches, il s'agit d'une entreprise de formalisation d'un type de calculateur, basé sur une distribution sur l'espace de variables continues, dont l'évolution dans le temps est une mise à jour basée sur la lecture de l'état de cellules voisines. Un des intérêts de la formalisation de ces systèmes est que l'on peut alors en étudier des propriétés telles que la stabilité, ce qui a été fait notamment par Angela Slavova [Slavova, 2000]. Elle a d'ailleurs publié des recueils d'articles, auxquels nous renvoyons le lecteur désireux d'approfondir la notion de CNN [Slavova, 2003; Slavova and Mladenov, 2004].

Pour nos recherches, dans la mesure où nos modèles ne rentrent pas exactement dans le canevas des CNN tel qu'il a été proposé dans [Chua and Yang, 1988], nous ne pouvons tirer parti des études de stabilité mentionnées ci-avant. Toutefois, la promotion de systèmes de calculs continus répartis sur un espace est une idée que nous soutenons également. Bien qu'actuellement nos modèles sont simulés sur ordinateurs, les collaborations que nous entretenons avec Bernard Girau [Girau, 2007] visent à en étudier la faisabilité sur VLSI, à l'instar de ce qui a été fait pour les CNN.

3.4 Modèles élémentaires

On trouve dans la littérature de nombreux systèmes dynamiques portant sur la modélisation d'une unité élémentaire de calcul. C'est ce que nous décrivons ici. L'intérêt de ces approches est que ces modèles peuvent être étudiés mathématiquement, ce qui devient plus délicat lorsque l'on considère un réseau de telles unités.

3.4.1 Dynamique du potentiel de membrane

Les travaux de modélisation réalisés par Hodgkin et Huxley [[Hodgkin and Huxley, 1952](#)] servent aujourd'hui de fondement à la description de l'activité neuronale par un système dynamique. En effet, Hodgkin et Huxley ont proposé un jeu d'équations différentielles pour décrire l'évolution de la différence de potentiel en un point d'un axone, entre l'intérieur et l'extérieur de la fibre. Nous renvoyons le lecteur au chapitre 2 de [[Gerstner and Kistler, 2002](#)] pour le détail du modèle, nous en retiendrons ici que la membrane est vue comme un condensateur. Les canaux ioniques sont des résistances quand ils laissent fuir les ions passivement, ou des sources de courant quand il s'agit de canaux actifs, qui forcent le passage de certains ions au travers de la membrane. Le potentiel de membrane, aux bornes du condensateur, est le fruit de la combinaison des effets des différents canaux. Le modèle est non linéaire en ce sens que les paramètres des canaux sont dépendants du potentiel courant du condensateur. La relation de dépendance, établie empiriquement, est complexe. Le potentiel est alors régi par une équation analogue à l'équation 3.1 page 38.

Comme ce modèle décrit une petite portion de fibre nerveuse, la simulation de la propagation le long de la fibre est réalisée en couplant plusieurs de ces portions, ce que l'on appelle des *modèles compartimentaux*. Le couplage entre deux compartiments voisins qui déroulent l'équation d'Hodgkin-Huxley se fait par les courants, qui peuvent, en plus de charger le condensateur, se diffuser d'un compartiment à ses voisins.

L'étude de l'équation d'une fibre est, en suivant ce modèle, extrêmement délicate, et peut-être résolue numériquement via des logiciels comme NEURON [[NEURON, 2008](#)], qui se basent sur ce modèle compartimental. Toutefois, afin de mieux analyser la dynamique du modèle d'Hodgkin-Huxley, une version simplifiée en a été proposée, sous la forme de l'équation de FitzHugh-Nagumo [[Izhikevich and FitzHugh, 2006](#)], qui est une équation de réaction-diffusion avec une dimension d'espace (le long de la fibre), comme dans le cas du modèle compartimental. Le principe, comme expliqué dans [[Gerstner and Kistler, 2002](#)], consiste à séparer des termes évoluant rapidement de ceux évoluant lentement dans le modèle d'Hodgkin-Huxley, et considérer que les termes rapides trouvent leur équilibre en un temps négligeable, durant lequel les termes lents peuvent être supposés constants. Cet équilibre, dépendant des termes lents, est réinjecté dans les équations du modèle, ne dépendant alors plus que des termes lents. D'autres modèles que celui de FitzHugh-Nagumo ont été proposés, et nous renvoyons le lecteur à [[Gerstner and Kistler, 2002](#); [Izhikevich, 2004](#)] pour plus de références sur ces modèles.

La finalité de ces modèles est généralement d'observer le résultat de leur dynamique, et de lier leurs paramètres, relatifs à des grandeurs biologiques, à la dynamique. Les simulations s'effectuent par des outils dédiés, comme NEURON que nous avons mentionné, mais aussi par le recours aux CNN (cf. paragraphe 3.3.3) [[Slavova and Zecca, 2003](#); [Slavova and Zecca, 2007](#)]. En comparaison avec la finalité de nos recherches, plus orientée vers les effets de population d'un réseau de calculateurs, ces études concernent un grain de modélisation biologique bien plus fin, puisqu'il est intra-neuronale, mais aussi bien moins « computationnel », puisque ces modèles ne sont pas analysés en terme de mécanismes de traitement de l'information. On peut toutefois

nuancer cette remarque par les travaux relatifs aux propriétés de transmission de l'information par des *burst*²² [Izhikevich, 2002; Izhikevich *et al.*, 2003], bien que leur portée reste cependant limitée à la synapse, ce qui est une échelle encore assez fine par rapport à celle que nous avons choisie.

3.4.2 Dynamique de population

La surface corticale possède une structure régulière, faite de neurones pyramidaux dont l'activité est régulée par des interneurons. Il en a été produit des modèles globaux, calculant l'activation d'une population de neurones pyramidaux, modélisant l'effet global de la population d'interneurons sur cette activation.

Parmi ces modèles, dits de *neural mass*, celui de Jansen et collègues [Jansen *et al.*, 1995; Jansen and Rit, 1995] permet de retrouver les oscillations électriques du tissu cortical, comme on les observe, par exemple, via un EEG²³. Le modèle décrit l'évolution de quelques paramètres (potentiels, sorties de synapses, etc...) en fonction de lois temporelles issues de données biologiques, et cette évolution, fortement non linéaire, ramène l'étude du modèle au cas de l'équation 3.1 page 38. Grimbert et Faugeras ont alors montré qu'un paramètre, modélisant l'activité extra-corticale reçue par le tissu, pouvait faire changer la dynamique du système, et ont mené une étude des bifurcations du modèle de Jansen [Grimbert and Faugeras, 2006]. Les auteurs concluent que le modèle est très sensible aux paramètres physiologiques, et qu'il est difficile d'étudier mathématiquement un réseau de tels systèmes dynamiques. L'étude de populations restreinte de ces modèles, basée sur des expérimentations, a été toutefois réalisée, comme par exemple dans [David and Friston, 2003] où les auteurs montrent que le changement de constantes de temps dans le modèle permet de retrouver tout le spectre EEG. On trouve enfin, dans la littérature récente, des raffinements de ce modèle, dont le but est, entre autre, d'étudier les effets de médicaments [Moran *et al.*, 2007].

Le modèle de Jansen est donc un système dynamique visant à coller finement aux observations EEG, obtenues dans le modèle comme effet de la dynamique du système. On peut alors identifier le rôle de différents paramètres, dont le modèle propose un lien avec des constantes biologiques. Il existe d'autres travaux [Lücke, 2005; Lücke and Bouecke, 2005a], basés de la même façon sur une analyse en population, qui modélisent une maxicouche corticale, en décrivant globalement l'activation de la population de minicouche qui la composent. Ces travaux sont, à l'instar des nôtres, plus orientés sur les propriétés computationnelles du modèle que sur sa capacité à « coller » quantitativement aux données biologiques. En particulier, l'analyse du champ de vecteur de l'espace des phases du modèle montre l'existence de plusieurs attracteurs, ce qui permet de discuter les propriétés de *symmetry breaking*. En effet, et nous y reviendrons au paragraphe 3.6.2, le fait que, dans une population d'unités de calcul soumises aux mêmes conditions, les comportements puissent malgré tout se séparer en plusieurs types, est une propriété de calcul essentielle, pour les prises de décision par exemple.

3.4.3 Quid de la modélisation fine de la dynamique biologique ?

Qu'il s'agisse du niveau du neurone ou de celui d'une population, les résultats que nous avons cités sur les études dynamiques prennent leur pertinence dans le cadre d'une modélisation précise d'observations biologiques. Pour ces systèmes, comme nous l'avons mentionné, il reste

22. Train de potentiels d'action très serrés de courte durée.

23. Electro-encéphalogramme.

difficile de mener une étude mathématique d'un réseau, la dynamique n'étant alors analysée que pour une seule unité de calcul (qui peut représenter toutefois une population).

Il s'avère que les systèmes que nous proposons ont justement la forme d'un réseau d'unités de ce type, mais contrairement aux objectifs des études que nous venons de présenter, nous nous intéressons aux propriétés de calcul de ces systèmes, plus qu'à leur capacité à s'accorder à des données expérimentales. Il serait donc pour nous frustrant de s'interdire la simulation de réseaux d'unités, bien que ceci implique, dans l'état actuel de nos connaissances, de se passer de la garantie que peuvent offrir des résultats mathématiques.

3.4.4 Modèles biologiques computationnels

Les modèles que nous avons présentés dans les paragraphes précédents se prêtent difficilement, comme nous l'avons mentionné, à une mise en réseau. Il existe en revanche des approches où ces modèles ont été simplifiés, afin de mener des études plus axées sur les propriétés calculatoires de réseaux d'unités inspirés de la biologie. Ainsi, la dynamique complexe des modèles de neurones a été depuis de nombreuses années résumée par des modèles comme les modèles *integrate and fire* [Knight, 1972] et *SRM0* [Gerstner, 1995]. Ces modèles ont extrait de la dynamique complexe de modèles comme ceux d'Hodgkin et Huxley l'essentiel de leurs propriétés spatio-temporelles. Le bénéfice de cette simplification est la possibilité de réaliser des simulations de populations, au sein desquelles apparaissent des dynamiques de groupe.

Nous renvoyons le lecteur désireux de connaître en détail ces modèles à [Gerstner and Kistler, 2002], nous n'en retiendrons ici que l'existence de *modèles de neurones événementiels*. Ces neurones émettent des *spikes*, dits aussi *potentiels d'actions*, que la modélisation réduit à des impulsions de Dirac localisées dans le temps. L'activité d'un neurone est alors l'intégration, dans le temps, des potentiels d'action que lui ont envoyés les neurones auxquels il est connecté. Ce type de simplification, à des fins calculatoires, est à rapprocher des motivations de la définition du neurone formel [McCulloch and Pitts, 1943], qui lui aussi réduit la dynamique du neurone à une fonction mathématique, en l'occurrence une séparation linéaire. Dans ce dernier cas toutefois, le caractère temporel du fonctionnement du neurone n'est pas pris en compte.

3.5 Systèmes dynamiques comme modèle de mémoire temporelle

Les systèmes dynamiques, comme leur nom le suggère, mais surtout du fait de leur formalisation par une équation différentielle en temps (cf. équation 3.2 page 38), ont la caractéristique de modéliser des phénomènes temporels. Cette lapalissade en appelle une autre, à savoir le caractère statique de la mémoire en informatique, vu que celle-ci est usuellement dédié à un stockage à long terme. Que ce soient les bases de faits et de règles en intelligence artificielle symbolique, les poids d'un perceptron multicouches dans le domaine de l'apprentissage statistique, ou une base de données dans un système informatique plus généralement, le stockage en mémoire des ordinateurs concerne majoritairement des représentations statiques, et il ne nous est pas naturel de *figer* dans la mémoire des ordinateurs des représentations temporelles. Dans la mesure où ce qui est figé dans un système dynamique est son équation de champ de vecteurs, c'est-à-dire la loi qui régit son évolution dans le temps, ils sont candidats à la représentation du temps sur nos machines de Turing.

3.5.1 Principe

Le principe de l'utilisation des systèmes dynamiques comme mémoire temporelle est celui du filtrage récursif utilisé en traitement du signal. Reste toutefois à s'assurer que ce que calcule le filtre a de bonnes propriétés. En effet, un filtre qui sature et reste indéfiniment dans cet état véhicule une représentation frustrée : « *Un événement, dans le passé, a causé une saturation* ». On ne peut rien savoir d'autre sur cet événement (sa date, son intensité, etc.). Les systèmes dynamiques étant des filtres récursifs dont, dans notre contexte, nous attendons de bonnes capacités de représentation temporelle, il convient d'être plus attentifs aux propriétés qui peuvent leur assurer une puissance de calcul intéressante. Ces propriétés sont présentées plus en détail dans [Legenstein and Maass, 2007b], seuls les grands principes sont esquissés ici.

Une étude sur les automates cellulaires²⁴ a montré que certains d'entre eux étaient complètement chaotiques, alors que d'autres conduisaient inexorablement à des états fixes ou des cycles simples [Langton, 1990]. Les plus intéressants, d'un point de vue calculatoire, sont ceux qui expriment une variabilité « *suffisante* » sans conduire à un chaos désorganisé. Langton a caractérisé les automates cellulaires par un paramètre, dont la variation permettait de passer d'automates conduisant à des états simple vers des automates chaotiques, ce qu'il compare à une transition de phase. Il a constaté que les automates intéressants sont ceux obtenus en limite de transition, juste avant le chaos. De façon plus générale, on considère que les systèmes dynamiques à la limite du chaos (*Edge of chaos*) sont les plus intéressants d'un point de vue calculatoire. Cette optimalité a été étudiée dans le cadre des automates cellulaires par une approche à base d'algorithmes génétiques [Packard, 1988], mais les résultats de cette étude par évolution simulée sont contestés [Mitchell *et al.*, 1993].

Toutefois, certains auteurs avancent [Legenstein and Maass, 2007b] que les systèmes biologiques s'auto-régulent pour rester en limite de chaos, ce que l'on nomme *Self-Organized Criticality*. Ce maintien actif à la frontière du chaos est une réponse, à notre sens pertinente, au fait que les systèmes dynamiques de ce type sont rares, en ce sens qu'un système dynamique dont la structure est aléatoire a peu de chance d'être de ce type.

Sortons du contexte des filtres récursifs pour proposer une architecture de calcul universelle [Maass *et al.*, 2002], comme l'est la machine de Turing, dont nous présentons les propriétés. Cette machine universelle, est constituée d'un état, qui est un système dynamique récurrent. Cet état est influencé en permanence par des entrées (cf. équation 3.2), et est lu par des sorties. Le système dynamique au cœur de la machine doit vérifier la propriété de *fading memory*, qui dit que la relation de dépendance entre l'état actuel et une entrée passée diminue avec le temps qui les sépare. Cette propriété peut ne pas être vérifiée pour un système chaotique, par exemple. Il en résulte que l'état courant tient compte des perturbations, i.e. des entrées, relativement récentes uniquement. La seconde propriété que doit vérifier la machine s'exprime au niveau des composants qui lisent l'état interne pour constituer la sortie. Ces composants doivent pouvoir *séparer* des états internes correspondant à des historiques différents.

Dans la littérature, l'état interne est comparé à un liquide avec viscosité, et les entrées à des perturbations de la surface de ce liquide. L'état de cette surface, à un moment donné, est représentatif de l'histoire des perturbations récentes, la viscosité assurant la propriété de *fading memory*. On parle alors de *Liquid computing* ou *Reservoir computing*. Comme ne le suggère pas l'abondance de la littérature sur ces modèles lorsque l'état liquide est un réseau de neurones impulsifs, cet état n'est pas nécessairement constitué de cette façon. Aussi mentionnerons-nous des travaux impliquant de la mémoire temporelle selon ce principe sans que ceux-ci n'impliquent des neurones impulsifs. C'est d'ailleurs la voie sur laquelle notre projet

24. Voir définition au paragraphe 3.3.

de recherche est engagé. Le lecteur pourra trouver une vue d'ensemble complète des systèmes neuronnaires récurrents de type *reservoir* dans [Lukoševičius and Jaeger, 2009].

3.5.2 Modèles impulsionsnels

L'opposition entre le codage fréquentiel, qui considère que les neurones s'échangent des nombres codés par la fréquence des *spikes*, et le codage temporel, qui soutient que c'est la date d'émission de chaque potentiel d'action qui porte l'information, anime encore aujourd'hui la communauté connexionniste. Il est clair que dans le contexte de la constitution d'une mémoire temporelle par un système dynamique, les tenants du codage temporel ont eu matière à proposer des modèles basés sur des neurones impulsionsnels (*spiking neurons*). Ainsi sont apparues des machines à états liquides dont l'état interne récurrent est un système dynamique constitué d'un réseau de neurones impulsionsnels. Les *Liquid State Machines* [Maass and Markram, 2004] sont de cette nature, ainsi que les *Echo State Machines* [Jaeger and Haas, 2004] qui leur sont très similaires (cf. figure 3.4).

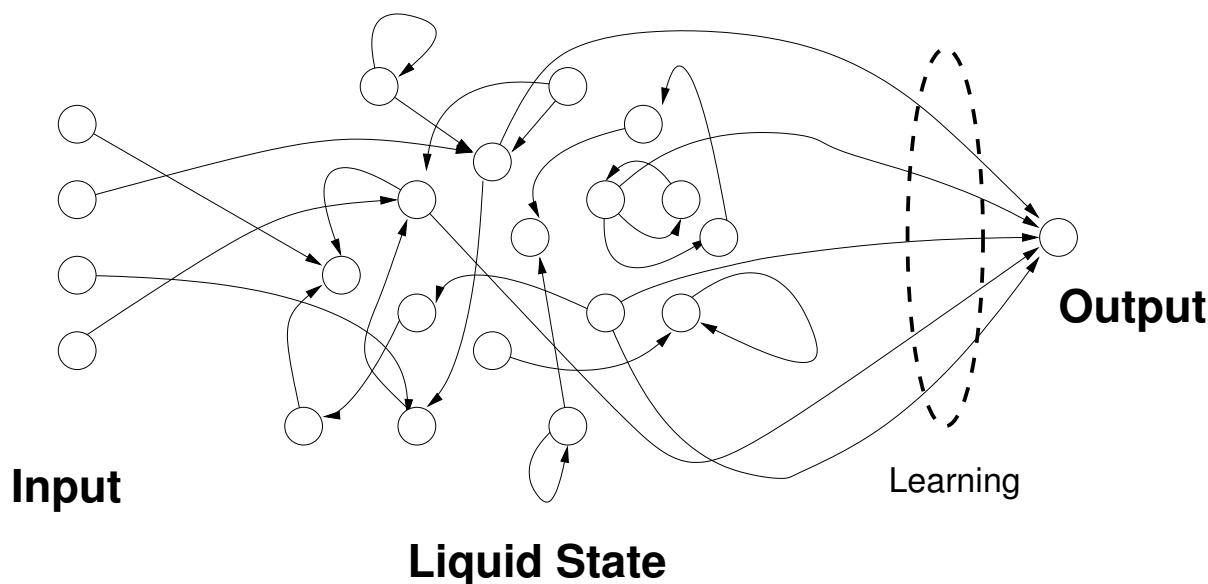


FIGURE 3.4 – *Liquid State Machine*. L'état est un réseau de neurones impulsionsnels, lus par un (ou des) neurone(s) de sortie. La fonction de lecture est usuellement un perceptron monocouche, qui peut être entraîné par apprentissage supervisé. Dans le cas d'une *Echo State Machine*, il y a un retour de la sortie vers l'état liquide.

Ces études ont en général vocation à s'inspirer du cortex, avec des neurones parfois plus complexes que SRM0 ou les intégrateurs à fuite, comme ceux proposés par Izhikevich [Izhikevich, 2006], et pour lesquels nous avons mentionné au paragraphe 3.4.2 la difficulté de l'étude mathématique du système complet. La fonction de lecture, qui construit des activités de sortie intègre les potentiels d'action de l'état interne, et est en général un séparateur linéaire. Les auteurs justifient ce choix simple par la haute dimensionnalité de l'état interne, invoquant des arguments analogues à ceux avancés pour les machines à vecteurs supports [Shawe-Taylor and Cristianini, 2000], sans pour autant toujours discuter le risque de malédiction de dimensionnalité auquel les machines à vecteurs supports savent, elles, résister. Mentionnons enfin des études relatives à des mesures de qualité du réseau constituant l'état interne [Legenstein and Maass,

2007a], pertinentes du fait de la difficulté de trouver un système dynamique à la frontière du chaos. Toutefois, l'idée qui est souvent retenue est de construire un réseau aléatoire faiblement connecté.

Enfin, d'autres auteurs proposent de simplifier la machine en utilisant comme état interne des attracteurs étranges (cf. paragraphe 3.1.2), ce qui revient à franchir la frontière du chaos. Par exemple, Crook propose comme état interne deux neurones, dont l'un est un oscillateur, et l'autre implémente une équation similaire à celle de l'attracteur de Rössler [Crook, 2007]. Ce modèle, quoiqu'ayant un état interne réduit, a toujours un formalisme impulsif.

3.5.3 Modèles fréquentiels

Nous souhaitons montrer dans ce paragraphe que les modèles à codage fréquentiel sont eux-aussi candidats à servir de base pour la construction d'un système dynamique pour les représentations temporelles. La plupart de ces modèles exploitaient des notions qui ont par la suite été reprises dans la formalisation des machines à état liquide. Par exemple, sur la base du perceptron multi-couches, l'on peut réaliser un traitement de séquences temporelles en introduisant des lignes à retard [Jordan, 1986; Elman, 1990]. Ainsi, si le signal d'entrée est un vecteur échantillonné à des instants successifs, on alimente le perceptron avec un vecteur d'entrée constitué de l'échantillon, mais aussi d'une copie, retardée d'un pas de temps, de la sortie ou de la couche cachée du perceptron. On construit ainsi un système dynamique, alimenté par un flux d'entrée, qui réagira à ce flux en fonction d'un état interne. Comme la sortie est un neurone formel, ces modèles, pourtant anciens, sont très proches des *Liquid State Machine*, alors qu'ils n'ont rien d'impulsif puisque les unités qui les composent sont des neurones formels « classiques ». D'autres approches ont également fait appel à ce que l'on qualifierait aujourd'hui de *Liquid State Machine*, sans que cela ait été formulé de la sorte. Nous citerons par exemple les travaux de modélisation de Dominey concernant les boucles cortico-striato-thalamiques impliquant le cortex frontal [Dominey, 1995]. Dans ce modèle, on trouve une structure composée de neurones ayant des réponses plus ou moins retardées, rappelant ce que d'autres auteurs appellent le *Spectral Timing* [Grossberg and Schmajuk, 1987; Grossberg and Schmajuk, 1989]. Cette batterie de neurones à retards, connectés aléatoirement entre eux, forment un système dynamique, auquel sont connectées des unités qui lisent cet état. Comme cet état est porteur de l'historique du système, les neurones de lecture peuvent *apprendre* à associer un état du système à un contexte pour prendre une décision d'action, ce qui permet de dire que ce modèle implémente une machine à état liquide, alors même que ce point est plutôt périphérique dans les travaux de Dominey.

Durant la même période, Dauce et collègues ont étudié la dynamique d'un réseau de neurones aux connexions asymétriques [Dauce et al., 1998]. Ce réseau est chaotique en l'absence d'entrées, mais stabilisé sur un attracteur lorsque des entrées sont maintenues, l'attracteur étant spécifique à la configuration des entrées. La reconnaissance s'effectue en comparant l'attracteur courant avec un attracteur de référence, ce qui est à rapprocher de ce que nous avons mentionné au paragraphe 3.2 concernant la modélisation du robinet qui fuit. Ces travaux ne visent certes pas à constituer une mémoire temporelle, mais nous les citons ici dans la mesure où l'« état liquide » qui y est présenté, sur la base de neurones à codage fréquentiel, a toutes les propriétés requises, et que l'analyse de l'effet des entrées comme une réduction de la dynamique du système nous paraît très pertinente.

Enfin, certains auteurs, comme Crook et collègues, dont nous avons mentionné des travaux au paragraphe 3.5.2, ont simplifié des modèles impulsifs de machine à état liquide pour proposer des modèles sans neurones du tout, où l'état liquide est un simple attracteur de Lorenz

(cf. paragraphe 3.1.2), lu par des observateurs qui échantillonnent dans le temps la sortie du système dans avoir recours à des neurones impulsifs [Goh and Crook, 2007].

3.6 Systèmes dynamiques comme modèle de mémoire spatiale

Nous avons, dans le paragraphe précédent, montré comment l'état instantané d'un système dynamique est porteur de l'historique des entrées qu'il reçoit, et comment on peut utiliser cette propriété pour traiter de représentations temporelles. Dans le présent paragraphe, nous interprétons la dynamique des systèmes un peu différemment, en mettant non plus l'accent sur la suite d'états exhibés par le système mais sur les états stables vers lesquels de telles suites convergent, quand elles convergent bien sûr.

Si l'on s'intéresse aux états stables d'un système, l'orbite qui conduit à l'état stable n'a pas d'autre intérêt que celui de ramener le système en cet état, et l'on ne s'intéressera pas à la suite des états parcourus, contrairement au paragraphe précédent où c'est justement sur la diversité des états d'une trajectoire que repose la puissance d'expression temporelle de ces systèmes. Nous parlerons alors, pour décrire l'évolution d'un système dont on ne s'intéresse qu'aux états stables, de *relaxation*. Nous nous focalisons dans ce paragraphe sur les systèmes de type connexionnistes, c'est-à-dire ceux dont la structure est un réseau d'unités (automates cellulaires, *cellular neural networks*, etc.). Un état, pour ces systèmes, est une configuration des activations des unités, et un état stable est une configuration qui se maintient dans le temps, puisqu'elle est conservée par application de l'équation d'évolution du système.

Nous donnerons ici la sémantique suivante aux états stables des systèmes dynamiques connexionnistes : un état stable exprime la *cohérence spatiale* des activités des unités du système. L'espace dont nous parlons, dans ce cas, est *l'espace d'états du système dynamique*. Ainsi, la relaxation d'un système connexionniste consiste, si l'on utilise cette sémantique, en une *restitution de cohérence spatiale* au sein du réseau d'unités. C'est en cela que, pour nous, ces systèmes sont le siège d'une mémoire spatiale, puisque c'est l'équation d'évolution du système qui définit, implicitement, ses états stables, et par conséquent ce que sont les activités cohérentes. Pour que cette sémantique ait un sens, il faut bien entendu que l'équation du système soit bien choisie, ou alors qu'elle soit le fruit d'un apprentissage visant à rendre cohérentes des activités qui doivent l'être.

Un exemple de tels systèmes est l'algorithme *Value Iteration* utilisé en *apprentissage par renforcement* dans le cas où la statistique du monde est connue [Sutton and Barto, 1998]. Les éléments de l'espace d'états S du problème de renforcement jouent ici le rôle d'unités, et leur activation est censée exprimer l'espérance de profit de la politique optimale. Cet espérance de profit, pour un état, dépend de celle des états que l'on peut atteindre depuis cet état, ce qui lie les espérances les unes aux autres²⁵. L'algorithme *Value Iteration* correspond à un système dynamique, pour lequel on peut réaliser une mise à jour asynchrone (cf. paragraphe 3.3.2), et dont on montre que le seul état stable est la distribution, sur les éléments de S , de valeurs qui correspondent justement à l'espérance de profit de la politique optimale. Ainsi, l'algorithme *Value Iteration* est la relaxation d'un système dynamique vers un état cohérent, solution du problème de calcul des espérances de profit de la politique optimale. Il s'agit bien d'une mémoire spatiale en ce sens que, dans l'équation de Bellman qui régit l'évolution du système, est implicitement stockée sur l'espace d'états cette distribution d'activités, que le système sait retrouver par relaxation.

25. C'est ce qu'exprime l'équation d'optimalité de Bellman.

Nous montrons dans ce paragraphe que cette approche est également intéressante pour des cas où le système a plusieurs états stables. Si l'on sait trouver une fonction de Lyapunov du système (cf. paragraphe 3.1.3), une configuration stable correspond à un minimum local de cette fonction, comme le rappelle la figure 3.2 page 41. Quand une fonction de Lyapunov existe, l'évolution du système revient à descendre ses pentes, comme nous l'avons vu. Notons que dans le cas d'une évaluation asynchrone d'un système connexionniste, le système ne restera pas en des points d'équilibre instable car tout état qui serait à la frontière de deux bassins d'attraction serait capté par l'un ou l'autre, en fonction de l'ordre d'évaluation des unités du système.

Pour notre projet de recherche qui, comme nous le verrons, s'appuie sur les champs de neurones dynamiques exposés ci-après, nous ne savons pas exhiber de fonctions de Lyapunov, mais le comportement du système dynamique peut être décrit qualitativement de la même façon, car le système relaxe vers le point d'équilibre le plus proche.

3.6.1 Modèle de Hopfield

Le plus classique des systèmes dynamiques ayant une mémoire spatiale est certainement le *réseau de Hopfield* [Hopfield, 1982]. Il s'agit d'un réseau d'unités à connectivité totale²⁶, de poids symétriques²⁷. Cette dernière contrainte permet d'exhiber une fonction de Lyapunov, et donc d'assurer que le réseau n'est pas pris dans des cycles. Pour les modèles de Hopfield, une mise à jour asynchrone est préférable [Hertz *et al.*, 1991]. L'unité choisie pour se mettre à jour calcule une somme, pondérée par les poids, des activités des autres unités et change son état booléen en conséquence.

Une procédure d'apprentissage préalable permet d'engrammer dans les poids du réseau des configurations d'activités, afin que celles-ci soient des attracteurs stables (en nombre limité toutefois). Ainsi, si l'on place le réseau dans un état donné en forçant celui de chacune de ses unités, on obtient, en laissant évoluer la dynamique asynchrone, un état qui correspond à l'un des attracteurs engrammés, comme l'illustre la figure 3.5. Selon notre sémantique, nous dirons que le réseau restitue la cohérence de l'activité en produisant l'état cohérent le plus proche, le réseau ayant été paramétré pour que les états cohérents soient ce que l'on souhaite y stocker. On parle aussi dans ce cas de *mémoire adressable par le contenu*.

Ce type de propriété a servi de base à la constitution de mémoires épisodiques artificielles, lors de travaux de modélisation fonctionnelle de l'hippocampe de l'Homme [Rougier, 2000; Rougier, 2001]. L'effet « madeleine de Proust » est en fait un exemple de mémoire adressable par le contenu, qui est en l'occurrence la madeleine. Le caractère spatial de cette mémoire est ici à prendre, comme nous l'avons dit, au sens de l'espace dans lequel se déploie le réseau, mais cette distribution spatiale d'activité peut coder tout type d'information.

3.6.2 Champs de neurones dynamiques

Les *champs de neurones dynamiques* (DNF²⁸) sont le type de systèmes dynamiques sur lesquels s'appuient nos travaux. Ils sont d'ailleurs inspirés pour la plupart par des observations biologiques concernant le tissu cortical. On trouvera dans [Simmering *et al.*, 2007] une bonne introduction aux DNF. Il s'agit de systèmes connexionnistes, mais dont la modélisation est réalisée sous forme d'un *continuum* d'unités. Ainsi, la majorité des travaux concernant les DNF

26. Chaque unité est connectée à toutes les autres.

27. La connexion de l'unité i vers l'unité j a le même poids que celle de j vers i .

28. *Dynamic Neural Fields*.

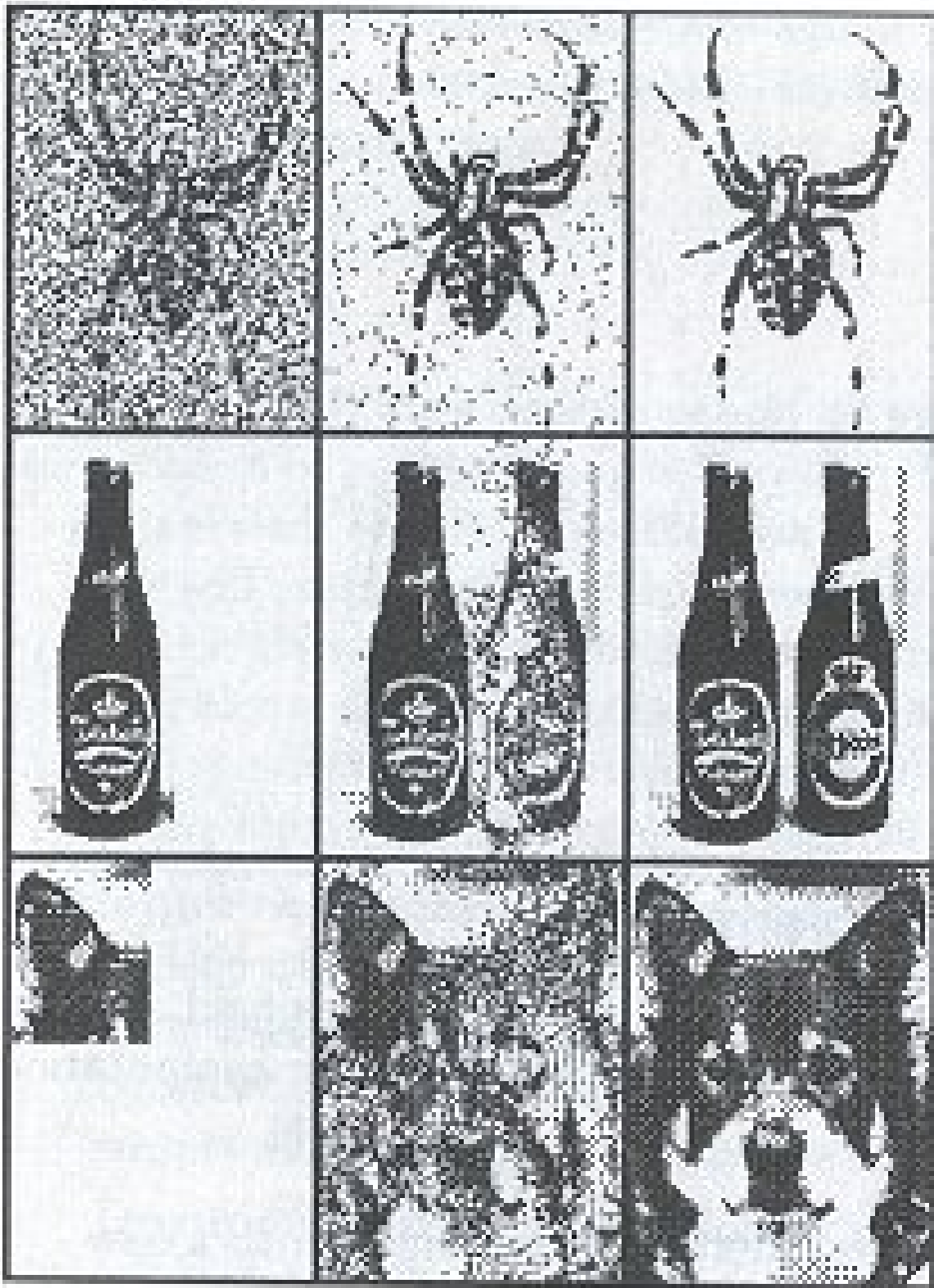


FIGURE 3.5 – Exemples de relaxations d'un même réseau de hopfield, à partir d'états bruités ou incomplets, d'après [Hertz *et al.*, 1991].

consistent à établir une équation sur l'espace continu des neurones, avec en général une justification biologique de la forme de cette équation. Est ensuite étudiée la dynamique de cette équation, stabilité, solutions périodiques sur l'espace, émission d'ondes, etc. Cette équation aux dérivées partielles régit l'évolution temporelle de chaque point du continuum (i.e. chaque unité) en fonction d'autres points du continuum, ce qui rapproche ces systèmes des équations de réaction diffusion.

Principe

La définition d'un DNF consiste tout d'abord à se donner un espace X d'unités, espace qui est continu et muni d'une distance. Cet espace est dans la majorité des études un espace 1D ou 2D. Une unité $x \in X$ de cet espace reçoit à chaque instant t une entrée $I(x, t) \in \mathbb{R}$. Cette entrée est imposée au DNF, et c'est au profil d'entrée qu'il réagit. Cette réaction consiste à calculer, en chaque x , un potentiel $u(x, t) \in \mathbb{R}$. L'équation « standard » d'un DNF est l'équation 3.6 proposée par Amari.

$$\tau \frac{\partial u(x, t)}{\partial t} = -\alpha u(x, t) + \int_{y \in X} w(x, y) f(u(y, t)) dy + I(x, t) + h \quad (3.6)$$

La fonction de *poids latéraux* $w(x, y) \in \mathbb{R}$ représente l'influence, positive ou négative, du potentiel de l'unité y sur l'unité x . En général, $u(y, t)$ est soumis à une non-linéarité f , de type sigmoïde ou fonction de Heaviside. Enfin, une constante $h \in \mathbb{R}$ est aussi introduite. Dans la littérature, les systèmes étudiés ont en général une fonction de poids homogène et radiale, et l'on remplace alors $w(x, y)$ par $w(|x - y|)$. Cette fonction est telle qu'elle est positive pour de faibles valeurs de $|x - y|$, négatives pour de grandes valeurs, puis s'annule en général à l'infini. C'est ce qu'on appelle une influence *on-center/off-surround*, très classique en biologie. En pratique, on utilise fréquemment des différences de gaussiennes, comme illustré sur la figure 3.6.

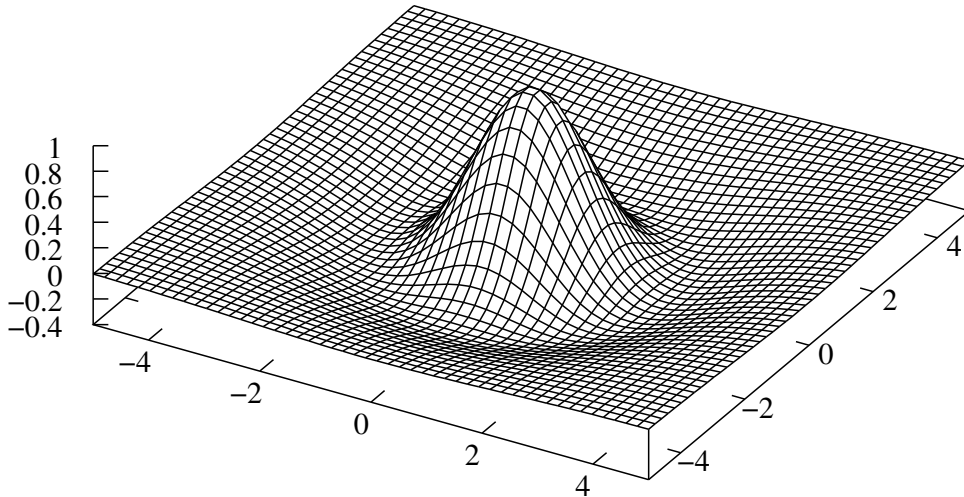


FIGURE 3.6 – La différence de gaussienne est utilisée pour modéliser le caractère *on-center/off-surround* de $w(|x - y|)$.

On peut montrer, pour le cas 1D [Amari, 1977] comme pour le cas 2D [Taylor, 1997], que selon les paramètres, il existe des solutions stables qui ont la forme d'activités u isolées sur l'espace, appelées « bulles²⁹ », ce que l'on peut vérifier en pratique, comme le montre notre simulation

29. On emploie le terme *bump* en anglais.

sur la figure 3.7. Ces résultats n'ont toutefois été établis mathématiquement que pour le cas où la distribution des entrées est constante. Ces travaux soulignent néanmoins l'importance de la forme de la fonction de poids et du paramètre h pour avoir la possibilité d'une formation de bulles d'activité. Notons aussi que sous certaines conditions, l'on peut voir apparaître des *instabilités de Turing*, comme cela arrive avec les équations de réaction diffusion. Il s'agit, pour ne pas rentrer dans les détails, d'états stables, qui forment des motifs tels des zébrures, ou les reliefs d'empreintes digitales. Le système converge vers l'une de ces formes, qui peut varier beaucoup avec le point d'où l'on part, mais qui est stable après convergence (cf. figure 3.8).

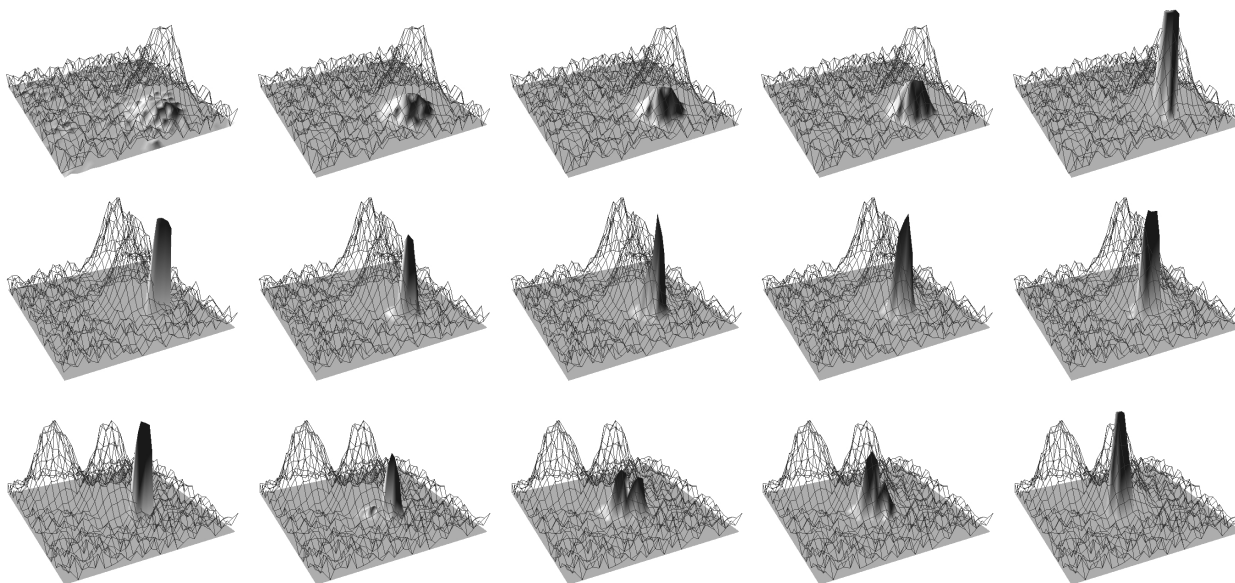


FIGURE 3.7 – Évolution (suivre le sens de lecture) du potentiel d'un champ de neurones dynamique. L'entrée I est affichée « en grille », le potentiel u en surface pleine. À chaque retour à la ligne au niveau des figures, la distribution de l'entrée est modifiée.

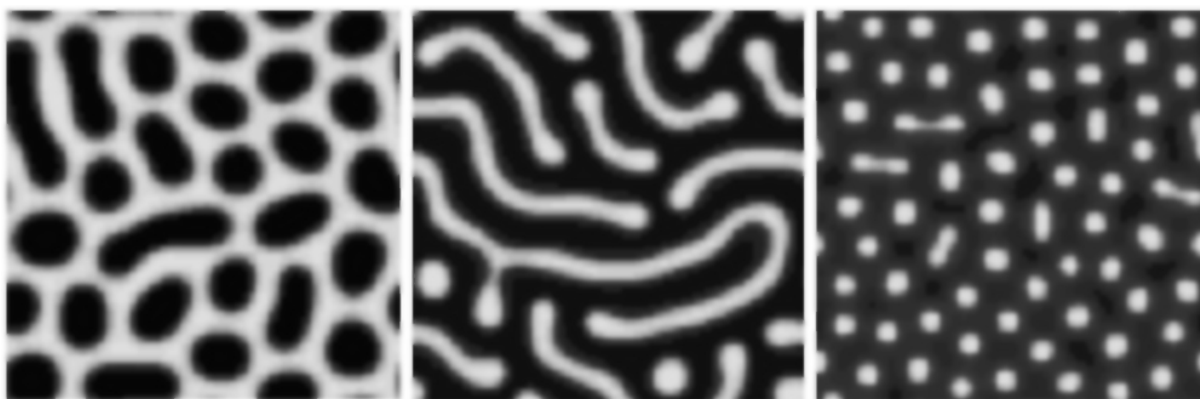


FIGURE 3.8 – Exemple d'instabilités de Turing. La surface de l'image est le champ de neurones, le potentiel u apparaît en niveaux de gris.

On trouve dans la littérature des conditions pour avoir une seule [Mikhailova and Goerick, 2005] ou plusieurs bulles [Murdock et al., 2006] dans le champ, ce qui peut être un point crucial

selon le contexte d'utilisation des DNF, nous y reviendrons. En pratique, lorsque le paramétrage permet la formation de bulles d'activité, ces bulles se placent là où la distribution $I(x)$ est *localement* la plus forte. Selon les paramètres toujours, une bulle peut s'auto-entretenir après s'être formée, même si l'entrée disparaît [Taylor, 1997]. Si le profil des entrées change, la nouvelle configuration de bulles, après stabilisation, est en général celle qui est la plus proche de la configuration avant le changement.

Ce système dynamique nous paraît intéressant si nous le considérons, comme c'est le cas pour tout ce paragraphe 3.6, comme un système qui conduit à une *restitution de cohérence* spatiale. En effet, les poids latéraux représentent une contrainte spatiale, que le champ « *s'échine à concilier* » avec le profil $I(x)$. Dans le cas d'un profil de poids latéraux comme celui de la figure 3.6, cette contrainte est que les activités doivent être des bulles isolées, ce que la dynamique « *restitue autant que faire se peut* » en plaçant les bulles d'activité comme on le voit sur la figure 3.7.

Nous avons essayé, lors du stage de DEA de Yann-Éric Douvier [Douvier, 2003], d'imposer d'autres profils, inhomogènes, et avons constaté empiriquement que la dynamique était beaucoup plus difficile à maîtriser. Il n'y a d'ailleurs à notre connaissance pas de résultats mathématiques pour ces cas-là.

Pour la simulation de ces champs enfin, les auteurs que nous mentionnons utilisent en général une résolution numérique de type Euler ou Runge-Kutta, dont le procédé est éloigné des systèmes à grain fin qui nous concernent. D'autres auteurs, dont nous sommes, simulent un système discrétisé par l'évaluation asynchrone [Ménard and Frezza-Buet, 2005; Rougier and Vitay, 2006] présentée au paragraphe 3.3.2. De plus, la connectivité latérale est assez dense, ce qui crée une occupation mémoire conséquente. Nous reviendrons sur ces points au paragraphe 3.8, mais nous soulignons ici que les difficultés informatiques liées à la simulation des DNF sont une des motivations des travaux que nous menons dans le cadre du projet `InterCell` présenté au paragraphe 8.1.2. D'autres solutions au coût calculatoire des DNF ont été proposées, au prix parfois d'un renoncement à une plausibilité biologique trop forte, pour résoudre ces problèmes. Dans [Rougier, 2006] par exemple, l'inhibition, plutôt que d'être véhiculée par des poids inhibiteurs très étalés (et donc nombreux) sur l'espace, se propage comme une onde via le champ excitateur.

Nous nous proposons dans les paragraphes qui suivent de considérer l'une des principales extensions de ce modèle, à savoir la prise en compte des temps de transmission de l'information dans le réseau. Nous présentons ensuite les DNF comme outil de traitement de l'information, ce qui correspond à l'utilisation que nous en faisons dans nos recherches pour ensuite discuter selon cette perspective de la nécessité d'une approche empirique des DNF afin de les exploiter dans des régimes où ils ne sont pas encore maîtrisés mathématiquement.

Prise en compte de délais de propagation

De nombreux travaux abordent l'extension du système dynamique en ajoutant des termes de délai de propagation dans l'équation 3.6. Ces travaux sont motivés par un souci de plausibilité biologique, l'étude du système dynamique obtenu permettant d'analyser des comportements d'oscillation et de propagation d'onde d'activité observés à la surface du cortex. Nous aborderons ces travaux sous l'angle des équations proposées et du comportement qualitatif des systèmes qui en résulte.

La première modification que nous aborderons consiste à définir pour chaque unité une deuxième activité, disons $v(x, t)$, qui augmente avec $u(x, t)$, mais de façon retardée. Cette activité inhibe u , simulant ainsi un phénomène d'épuisement du neurone. Le système dynamique est celui

définit par l'équation 3.7.

$$\begin{aligned}\tau \frac{\partial u(x, t)}{\partial t} &= -u(x, t) + \int_{y \in X} w(|x - y|) f(u(y, t)) dy + I(x) - \beta v(x, t) \\ \epsilon \frac{\partial v(x, t)}{\partial t} &= -v(x, t) + u(x, t)\end{aligned}\quad (3.7)$$

Dans le cas 1D les valeurs de paramètres qui autorisent l'existence d'une onde ont été étudiées mathématiquement [Bressloff and Folias, 2004; Folias and Bressloff, 2004], et les auteurs ont fait mention d'états oscillants (appelés *breathers*) qui peuvent rester stable (l'oscillation persiste), ou alors générer des ondes, la figure 3.9 en donne l'illustration. Les auteurs proposent également un modèle plus proche de celui d'Hodgkin-Huxley, montrant que l'on retrouve le même type de résultats avec des neurones impulsionnels. L'étude des *breathers* a été généralisée au cas bidimensionnel [Folias and Bressloff, 2005], et l'on peut trouver des démonstrations de ces cas dans [Folias, 2008].

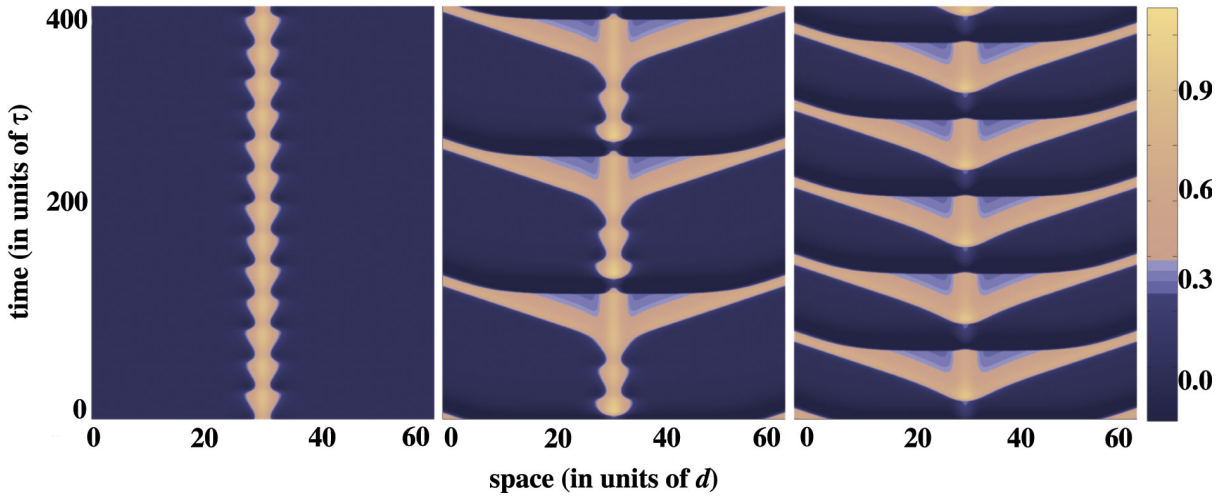


FIGURE 3.9 – Avec une entrée gaussienne, l'équation 3.7 génère un *breather* qui produit régulièrement une onde. De gauche à droite, c'est l'intensité de l'entrée présentée au centre du champ qui augmente. D'après [Folias, 2008].

Ce que l'on peut retenir de ce modèle, est qu'il met en œuvre en chaque point de l'espace un système d'équations couplées, qui est perturbé par les influences latérales et l'entrée. Le tracé des valeurs u et v de l'équation 3.7 pour une unité oscillante rappelle les orbites d'un système dynamique qui change de mode selon ces perturbations (cf. figure 3.10).

D'autres études, partant toujours de l'équation 3.6, tiennent compte du temps de propagation le long des connexions, en le supposant d'une vitesse constante v . Ainsi, le retard entre x et y est $\frac{|x-y|}{v}$. On supposera également une atténuation des signaux passés, avec une fonction η qui est une exponentielle décroissante. Ceci conduit à poser l'équation 3.8 étudiée dans [Enculescu and Bestehorn, 2007; Venkov et al., 2007], sachant qu'une forme voisine a également été étudiée dans [Hutt et al., 2003]. On notera que, contrairement à l'équation 3.6 d'Amari, les connexions latérales mettent à jour directement u , et non sa dérivée.

$$u(x, t) = \int_{-\infty}^t \eta(t - s) \int_{y \in X} w(|x - y|) f\left(u\left(y, s - \frac{|x - y|}{v}\right)\right) dy ds \quad (3.8)$$

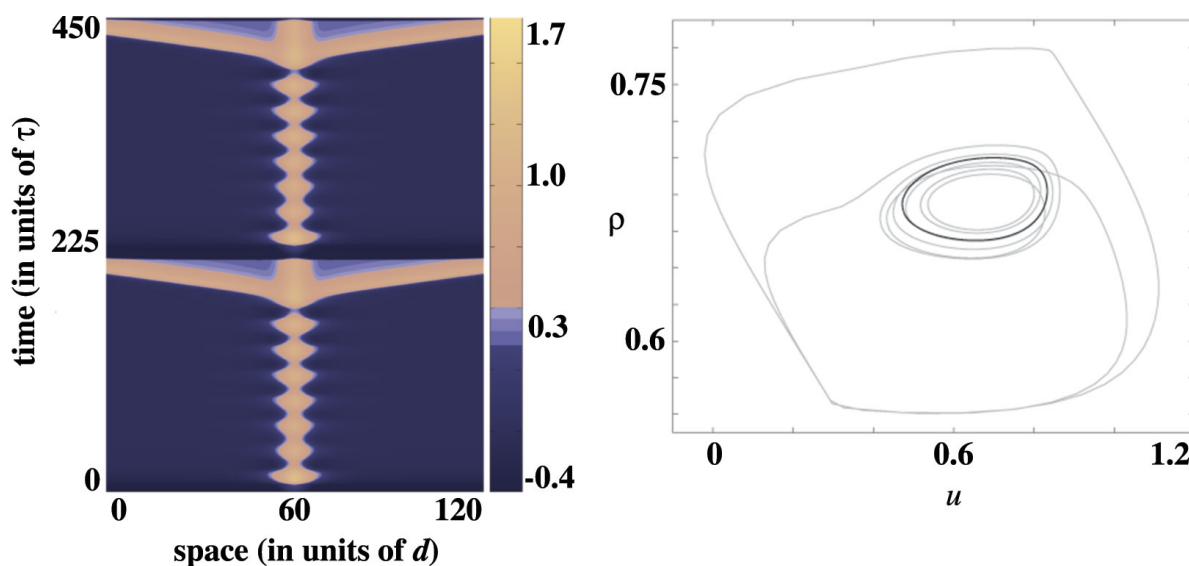


FIGURE 3.10 – Évolution des valeurs u et v pour l'unité centrale (v est noté ρ sur la figure) lorsque le système a le comportement illustré au centre de la figure 3.9. D'après [Folias, 2008].

La résolution de cette équation par une méthode de Runge-Kutta permet d'analyser les effets du paramètre de vitesse de propagation. On trouvera une étude analytique détaillée de ces effets dans [Venkov *et al.*, 2007], dont nous avons tiré la figure 3.11, sur laquelle on notera l'apparition d'une instabilité de Turing.

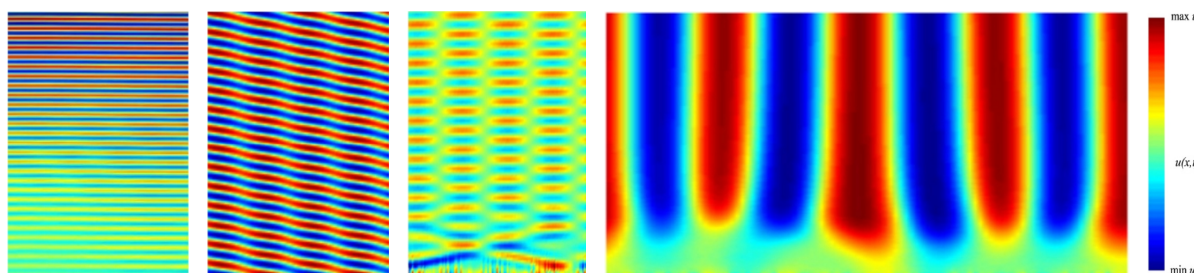


FIGURE 3.11 – Résolution de l'équation 3.8 dans des régimes différents. L'abscisse des figures est le champ 1D, l'ordonnée est le temps (de bas en haut). La deuxième figure en partant de la gauche correspond à une propagation d'onde, et la figure de droite est une instabilité de Turing. D'après [Venkov *et al.*, 2007].

Nous citerons enfin l'équation 3.9 proposée par Sabatini et collègues [Sabatini *et al.*, 2004; Sabatini *et al.*, 2005], qui contrairement à ce que nous avons vu précédemment, implémente un terme de diffusion sous la forme d'un Laplacien spatial. Cela dit, précisons que la discrétisation d'un Laplacien crée effectivement une influence on-center/off-surround, et que ceci est cohérent

avec ce que nous avons présenté jusqu'ici.

$$\begin{aligned}
 \tau \frac{\partial e(x,t)}{\partial t} &= -\alpha_e e(x,t) + D \frac{\partial^2 e(x,t)}{\partial x^2} \\
 &\quad -b(e(x-d,t) + e(x+d,t)) \\
 &\quad -w_{ie} f(i(x,t - \tau_{ie}) + s(x,t)) \\
 \tau \frac{\partial i(x,t)}{\partial t} &= -\alpha_i i(x,t) \\
 &\quad +w_{ei} f(e(x,t - \tau_{ei}))
 \end{aligned}
 \tag{3.9}$$

Dans ce cas, les auteurs proposent un réseau 1D, dont chaque unité est un couple de neurones, l'un excitateur et l'autre inhibiteur, formant ainsi un oscillateur illustré sur la figure 3.12. On aurait pu interpréter l'équation 3.8 de la même façon. La dynamique du système permet de coupler les oscillations des unités qui reçoivent en entrée des trains d'impulsions booléennes denses, formant alors une cohérence spatiale et temporelle. Si les entrées sont moins fréquentes, les oscillateurs se découplent, perdant leur cohérence spatio-temporelle. Pour la stimulation d'un seul neurone par une entrée constante, selon les paramètres, on peut retrouver des *breathers*. Ce modèle est assez proche d'un modèle impulsif.

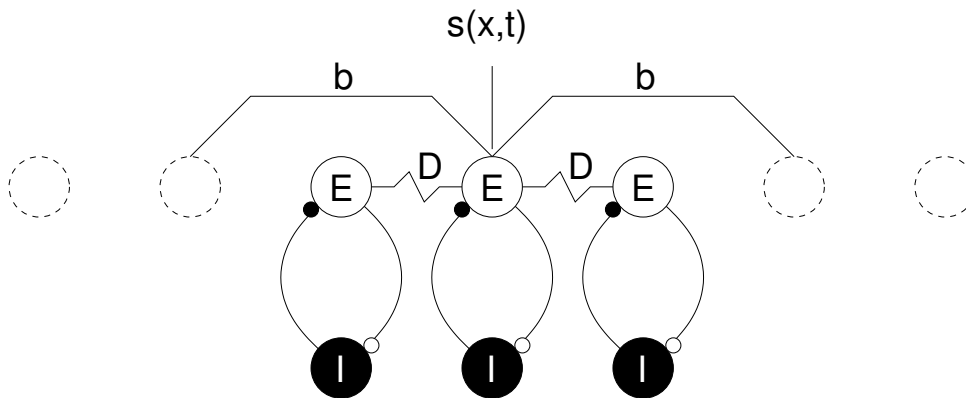


FIGURE 3.12 – L'équation 3.9 revient à coupler des oscillateurs pour constituer le champ de neurones. D'après [Sabatini *et al.*, 2004]

Pour résumer le survol que nous avons fait des équations de champs de neurones avec délais temporels, nous retiendrons que ces délais introduisent des oscillations, et que le couplage spatial permet sous certaines conditions l'existence d'ondes qui se propagent. Ces résultats sont obtenus par des études analytiques des équations et de la stabilité des systèmes dynamiques qu'elles constituent. Les simulations sont produites en général par des méthodes de Runge-Kutta, avec une distribution des entrées destinée uniquement à initier un mode de fonctionnement du système pour l'étude de sa dynamique.

Les champs de neurones comme outils de calcul

Notre perspective, comme nous l'avons dit au paragraphe 2.2, n'est pas de rendre compte d'observations biologiques par la production d'un modèle, contrairement à ce qui motive les travaux cités précédemment, mais à utiliser les DNF comme outil de traitement de l'information.

Nous avons mentionné page 57 que nous interprétons les DNF comme une technique permettant une *restitution de cohérence spatiale* à partir du profil d'entrée qui leur est imposé. Dans le

cas des connexions latérales *on-center/off-surround*, cela revient à isoler une activité autour des neurones qui reçoivent localement les entrées les plus fortes.

Nous considérons que, dans ce cas toujours, le champ de neurone réalise par une action collective une *réduction de l'information* présente dans le profil des entrées, et nous interprétons cette réduction d'information comme une décision collective. Ce mécanisme nous apparaît comme pouvant faire le lien, de façon bien plus subtile que ne le font les seuils, entre une donnée continue et le caractère nécessairement discret et tranché d'une décision à prendre sur cette donnée.

C'est en ce sens que les DNF sont de bons candidats pour modéliser les phénomènes d'attention. En effet, porter attention, c'est choisir parmi les stimuli. Comme rappelé dans [Mikhailova and Goerick, 2005; Douglas and Martin, 2007], la dynamique des bulles d'activité permet d'assurer des propriétés comme la compétition, la coopération, la correction, et la robustesse aux distracteurs. La figure 3.13 détaille ce propos.

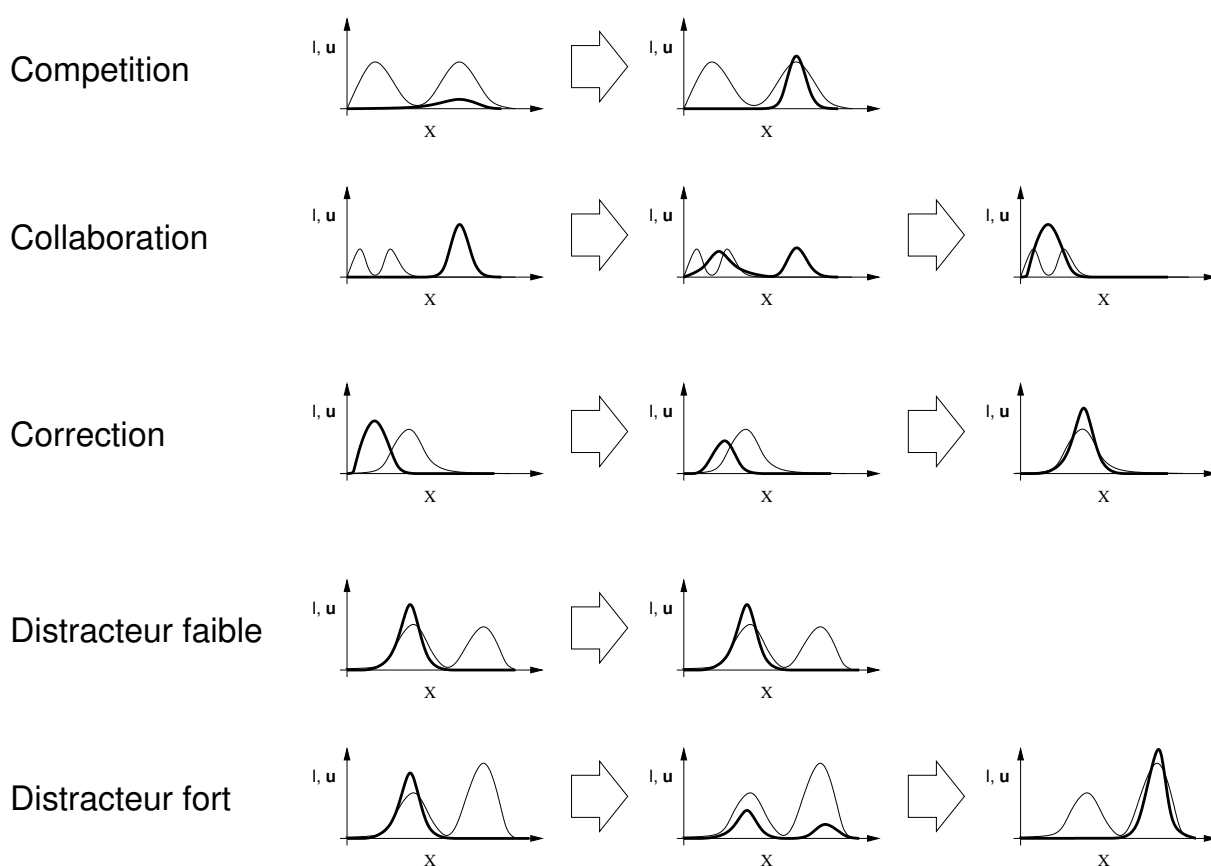


FIGURE 3.13 – L'évolution de u (en gras) en réponse à I réalise des opérations de compétition, de coopération (lorsque I est constitué de plusieurs activités regroupées) et de gestion des distracteurs. Ce type de dynamique s'observe également sur la figure 3.7. D'après [Mikhailova and Goerick, 2005].

Ces propriétés de calcul permettent par exemple de focaliser l'attention visuelle à partir d'un champ 2D, dont les entrées correspondent à des points de saillance sur l'image [Mikhailova and Goerick, 2005; Rougier and Vitay, 2006; Fix *et al.*, 2007; Simmering *et al.*, 2007], ce qui est également proposé comme modèle de l'activation du colliculus supérieur [Schneider and Erlhagen, 2002]. Dans ces travaux, il y a coopération entre plusieurs champs de neurones, ce qui rappelle les champs inhibiteurs exprimés par les variables auxiliaires v et i des équations 3.7 et 3.9. Dans

les études évoquées ici toutefois, la concordance avec une entrée est primordiale, les auteurs ne mettent pas en avant la capacité des champs qu'ils utilisent à générer des *breathers* ni des ondes.

Dans nos propres travaux [Ménard and Frezza-Buet, 2005], nous étendons cette notion de décision à un contexte multi-modal, ce qui crée une dynamique conjointe entre plusieurs champs. Nous discuterons ce point en détail au paragraphe 4.5.

Citons enfin une exploitation un peu spécifique de la dynamique des champs de neurones, dans le contexte de la planification de trajectoire en milieu contraint, par des obstacles par exemples [Menzner *et al.*, 2000; Iossifidis and Steinhage, 2001]. Le champ de neurones, dans ces travaux, représente l'espace dans lequel se réalise la trajectoire. L'entrée est positive à l'endroit du point à atteindre, et négative à l'endroit des obstacles. La bulle d'activité représente la position courante, et on peut en induire le mouvement en dissymétrisant le noyau de poids latéraux. La bulle se déplace ainsi vers le but, mais évite les obstacles. Des travaux un peu antérieurs en robotique située proposent d'avoir recours à cet usage des DNF dans les cas où la dynamique du robot est indéterminée, c'est-à-dire là où persiste des degrés de liberté non contraints par la tâche à réaliser [Bergener *et al.*, 1999]. La relation entre le mouvement des bulles et la dissymétrie du noyau de poids latéraux est complexe, et suppose en particulier qu'il n'y ait pas d'hystérésis au niveau des unités [Horta and Erlhagen, 2006].

Ces différents modèles, même s'ils n'empruntent pas les mêmes voies que les nôtres, sont en accord avec notre vision des champs de neurones dynamiques, car ils mettent en avant leur puissance de calcul bien plus que leur capacité à rendre compte d'observations biologiques.

3.6.3 Pertinence d'une étude empirique pour évaluer les champs de neurones

Comme nous l'avons affirmé, notre projet de recherche est concerné par les DNF du fait de leur capacité à proposer une activation qui accorde le profil des entrées à des contraintes exprimées dans les connexions du réseau. Or c'est ce point qui est, à notre connaissance, le moins abordé par les études mathématiques. Il nous apparaît alors nécessaire de proposer une méthodologie pour appréhender les propriétés de ces champs, basée, faute de mieux, sur des résultats empiriques.

Ce travail a été initié lors du stage de Master de Jérôme Carretero [Carretero, 2007]. Il fait suite à une première tentative [Frezza-Buet and Ménard, 2005] que nous avons développée pour qualifier le mécanisme de compétition utilisé dans le modèle *bi_jama*, présenté au paragraphe 4.5. Notre méthodologie consiste à proposer d'une part une batterie de critères numériques de qualification du profil u et de son accord avec i , et d'autre part une série d'expériences « *standard* » (i.e. une succession de profils i) auxquelles on peut soumettre le réseau. Les successions de profils d'entrée utilisés dans l'illustration de la figure 3.13 en sont un exemple possible. L'idée, à terme, est d'identifier des scénarios type, qui soient la réunion d'expériences et de mesures, qui permettent de qualifier la capacité d'une équation de DNF à résoudre tel ou tel type de problème.

Ce besoin est également exprimé par d'autres auteurs [Rougier and Vitay, 2006], qui s'inspire de [Deneve *et al.*, 1999] pour qualifier les propriétés des DNF à réaliser des traitements attentionnels.

Les travaux de thèse de Lucian Alecu ont permis de pousser plus avant la définition de critères numériques pour qualifier le « bon » comportement d'un champ. Ce critère numérique, défini dans [Alecu *et al.*, 2011], peut se résumer comme suit. L'activité en chacune des unités x

d'un champ neuronal à un instant t est une fonction $u(x, t) \in \mathbb{R}$. On définit de même l'entrée $i(x, t)$ à laquelle le champ doit répondre, en formant des bulles d'activité. Soit \mathcal{B} l'ensemble des activités du champ qui sont constituées de bulles éparses, les travaux de Lucian Alecu affirment que ce que doit calculer le champ est l'élément $u \in \mathcal{B}$ qui est le plus proche de la distribution i . De ce constat, il dérive un critère numérique, basé sur des distances, qui détermine dans quelle mesure l'activité courante $u(\cdot, t)$ du champ satisfait à cet objectif.

Durant ces travaux de thèse, nous avons montré expérimentalement que le critère numérique est cohérent avec les propriétés attentionnelles attendues des champs neuronaux (cf. figure 3.13), et que l'optimisation de ce critère permet de régler l'ajustement des paramètres du champ, problème toujours délicat avec ces systèmes dynamiques [Alecu and Frezza-Buet, 2009b]. Ce critère permet également de quantifier la capacité d'un champ à servir de processus compétitif dans une architecture d'auto-organisation, ce qui est l'apport principal de la thèse de Lucian Alecu. Nous présenterons ce point au paragraphe 4.5.4.

3.6.4 Bilan

Nous avons pris le parti de présenter les champs de neurones dynamiques comme un système dynamique restituant une cohérence spatiale entre ses unités, alors même que l'introduction de délais induit des comportements temporels du réseau, comme l'émission d'ondes qui se propagent sur le champ. Nous avons toutefois mis en avant les propriétés de calcul que l'on peut attendre de ces champs, contrairement à ce qui est fait dans la plupart des études que nous avons citées, et ces propriétés de calcul sont, elles, du côté de la restitution de cohérence qui nous préoccupe.

Que faire alors des dynamiques temporelles observées dans les champs de neurones avec délais? Nous prenons le parti de considérer l'émission des ondes, tout comme les effets de type *breather*, comme des effets de bord de la dynamique de ces champs, effets de bord que l'on observe en absence de profil d'entrée réaliste, vu que celui-ci est inexistant ou très simplifié dans ces études. D'ailleurs, le modèle de colliculus que nous avons mentionné [Schneider and Erlhagen, 2002] va dans ce sens, puisqu'il ne s'affranchi pas, lui, des entrées, et ne met pas en avant les phénomènes temporels alors que le champ a une structure similaire à l'équation 3.7.

Reste alors posée la question du rôle de ces délais. On peut par exemple proposer que l'inhibition agit comme un phénomène d'habituation, retardé, qui évite aux bulles d'activité de persévérer trop longtemps en s'auto-entretenant à un endroit du champ³⁰. Il s'agit alors d'un rôle fonctionnel bien plus fondamental, car la difficulté de contester par une nouvelle entrée une activation déjà établie sous forme d'une bulle dans un champ est un problème que nous rencontrons dans nos travaux. Qu'en l'absence d'entrées, cette inhibition retardée produise des phénomènes de propagation n'est alors plus vraiment une question centrale.

Nous souhaitons également ici nuancer l'intérêt d'un décodage des activités u de la population tel qu'il est proposé dans [Deneve et al., 1999]. En effet, à moins qu'il s'agisse d'effectuer une mesure empirique de ce que fait le champ, ou à moins qu'il s'agisse de convertir l'activité du champ en la commande d'un effecteur, la nécessité de décodage apparaît secondaire dans un système homogène où l'information est traitée par des champs neuronaux, comme celui que nous proposons (cf. paragraphe 4.5).

Nous pensons que notre recherche, comme d'autres, permet de rappeler avec la plus vive insistance qu'un champ de neurones est, que ce soit en informatique ou en biologie, un système dynamique complexe *couplé* avec d'autres ou avec l'extérieur, et qu'une étude d'un champ isolé est nécessairement incomplète.

30. Cette remarque m'a été faite par Nicolas Rougier.

3.7 Apprentissage et systèmes dynamiques

La notion d'apprentissage dans les systèmes dynamiques nécessite, pour être discutée, de prendre quelques précautions. En effet, n'importe quel apprentissage, de la part d'un système, est un changement d'état (modification des poids d'un perceptron par exemple). Peut-on dire pour autant que les équations de Lorenz sont des équations d'apprentissage ? Nous pensons qu'il ne s'agit que d'une question de point de vue, et c'est notre point de vue que nous précisons ici.

Repartons de l'équation 3.2 page 38, et modifions-la un peu, afin d'y faire apparaître un paramètre θ , ce qui donne l'équation 3.10.

$$\frac{dx}{dt} = X_{\theta}(x(t), e(t)) \quad (3.10)$$

C'est en jouant sur θ que l'on observe les bifurcations par exemple (cf. paragraphe 3.1.2). L'apprentissage dont nous parlerons consiste à modifier θ , et nous ne considérerons pas une modification de x comme un apprentissage. Ainsi, si pour un jeu de poids latéraux fixes d'un champ de neurones (i.e θ fixe), une entrée transitoire crée une bulle u qui persiste. Cette distribution de u est un état stable x si l'on prend les notations de l'équation 3.10. Ce profil u n'est pas considéré, pour la discussion qui suit, comme le résultat d'un apprentissage. Il n'en reste pas moins vrai que cet état, d'un point de vue fonctionnel, est le stockage d'un événement passé, et peut avoir comme fonction d'être un événement stocké, appris, et que notre restriction de la notion d'apprentissage à θ n'a que vocation à préciser ce dont nous parlons dans ces paragraphes suivants.

Nous étudions avec le modèle **bijama**, présenté au paragraphe 4.5, un système dynamique qui apprend en ligne, ce qui revient à dire que θ varie selon les états visités et l'entrée. Ces systèmes sont décrits en toute généralité par l'équation 3.11.

$$\begin{aligned} \frac{dx}{dt} &= X_{\theta(t)}(x(t), e(t)) \\ \frac{d\theta}{dt} &= X'(\theta(t), x(t), e(t)) \end{aligned} \quad (3.11)$$

Usuellement, on conçoit que X' est une règle d'apprentissage faisant évoluer θ assez lentement, au regard des modifications que subit x . Mais cela n'a rien d'obligatoire, et un tel système se ramène à un système dynamique usuel en remarquant que l'on retrouve l'équation 3.11 en remplaçant x dans l'équation 3.2 par $(x(t), \theta(t))$.

3.7.1 Techniques de *Machine Learning*

Comme il est souligné dans [Igel et al., 2001], « *There is a growing interest in using dynamic neural fields for modeling biological and technical systems, but constructive ways to set up such models are still missing* ». Il apparaît alors naturel, même si cela pose des difficultés, d'appliquer les techniques d'apprentissage automatique pour trouver le paramètre θ de l'équation 3.10. C'est ce qu'ont fait il y a plus de vingt ans Elman et Jordan, en détournant un perceptron multi-couches pour construire un système dynamique avec lignes à retard pour le traitement de séquences [Jordan, 1986; Elman, 1990], comme nous l'avons dit au paragraphe 3.5.3. Plus récemment, les techniques issues des travaux de Vapnik [Vapnik, 2000] (dont [Shawe-Taylor and Cristianini, 2000] donne une introduction plus abordable) ont également été appliqués aux champs de neurones dynamiques [Giese, 2003]. Ces techniques ont toutefois l'inconvénient de ne pas se prêter à un traitement en ligne.

Depuis les travaux d'Elman et Jordan, nous constatons que les techniques d'apprentissage automatique supervisé n'ont que peu influencé la conception de systèmes dynamiques neuro-mimétiques, et nous n'approfondirons pas plus dans la mesure où nous n'avons pas emprunté de voie similaire. On pourrait opposer à cette remarque l'utilisation d'apprentissage supervisé dans le cadre des neurones de sorties des machines à état liquide (cf. paragraphe 3.5.2), mais il nous apparaît que cette partie du modèle est justement celle qui n'exploite pas la notion de système dynamique.

3.7.2 Techniques évolutionnistes

Face au difficile problème de trouver le système dynamique adéquat pour une tâche donnée, le recours à des méthodes évolutionniste est envisageable. Il est clair toutefois que la définition du codage et des opérateurs de croisement est une injection forte de connaissance, et il est faux de considérer que la grande force de ces approches est de pouvoir trouver une solution « *en aveugle* ».

Dans le cadre de réseaux de neurones servant de supports à des systèmes situés, en interaction avec leur environnement, nous avons déjà mentionné les travaux de Randall Beer et collègues au paragraphe 1.3.1. Ces auteurs proposent de construire un système dynamique pour résoudre des tâches cognitives par un algorithme génétique [Slocum *et al.*, 2000; Beer, 2003]. Cette méthode rappelle d'ailleurs ce qui avait été proposé par Frédéric Gruau pour guider l'évolution des architectures neuronales afin de piloter un robot-fourmi [Gruau, 1993; Gruau, 1995], méthodes basées sur le principe de *programmation génétique*. Les opérateurs d'évolutions agissent non plus sur un code de l'individu, mais sur un programme (de type LISP) dont l'exécution génère l'individu. Les individus générés sont des réseaux de neurones aux poids modifiables, ce qui crée une dynamique évolutionniste complexe (voir la notion d'*Effet Baldwin* dans [Gruau, 1995]).

Enfin, notons aussi l'existence de travaux plus récents qui comparent des approches évolutionnistes à des approches à base de descente de gradient pour l'établissement des paramètres d'un champs de neurones [Igel *et al.*, 2001].

Nous avons choisi, pour notre part, de ne pas nous aventurer sur la voie des algorithmes évolutionnistes. En effet, les architectures que nous proposons sont le fruit d'une conception dirigée vers l'obtention de certains effets de population (auto-organisation conjointe, etc.). Bien que la voie évolutionniste, et en particulier dans le cas de la programmation génétique, puisse revendiquer une justification biologique, il s'agit surtout à l'heure actuelle d'une technique de recherche opérationnelle de force brute, de type recuit simulé [Premti, 1983], pour laquelle il faut avoir des a priori à injecter au niveau du codage des gènes. Nous préférons injecter nos a priori dans une architecture pour la simuler, ce qui nous paraît plus accessible.

3.7.3 Apprentissage Hebbien

L'apprentissage Hebbien est couramment employé dans les systèmes dynamiques connexionnistes car il est un modèle des modifications synaptiques des neurones réels, et les systèmes dynamiques que l'on trouve dans la littérature ont fréquemment vocation à modéliser la réalité biologique. Une règle de Hebb classique³¹ est par exemple utilisée dans le modèle de Hopfield que nous avons présenté au paragraphe 3.6.1, bien qu'une séparation stricte entre une phase d'engrammation des motifs, où cet apprentissage est mobilisé, et une phase de reconnaissance prête à discussion en terme de plausibilité.

31. dont les arguments sont les fréquences de décharge et non les instants d'arrivées des potentiels d'action.

L'apprentissage Hebbien est également impliqué dans [Dauce *et al.*, 1998] pour stabiliser les motifs d'entrée (voir paragraphe 3.5.3), ce qui étend le modèle de Hopfield aux cas de poids non-symétriques.

Toutefois, comme de nombreux modèles ont un caractère impulsionnel et non fréquentiel, c'est fréquemment une règle de type *STDP*³² qui est utilisée dans les modèles. Cette règle définit la valeur de la modification de poids occasionnée par la concomitance dans le temps d'un potentiel d'action pré et post-synaptique, les modifications de poids n'étant pas de même signe suivant la précedence de l'un ou de l'autre, comme l'illustre la figure 3.14. Nous renvoyons le lecteur à [Gerstner and Kistler, 2002] pour plus de détails sur cette règle, pour laquelle une équivalence avec une version étendue de la règle de Hebb fréquentielle³³ a été montrée [Izhikevich and Desai, 2003].

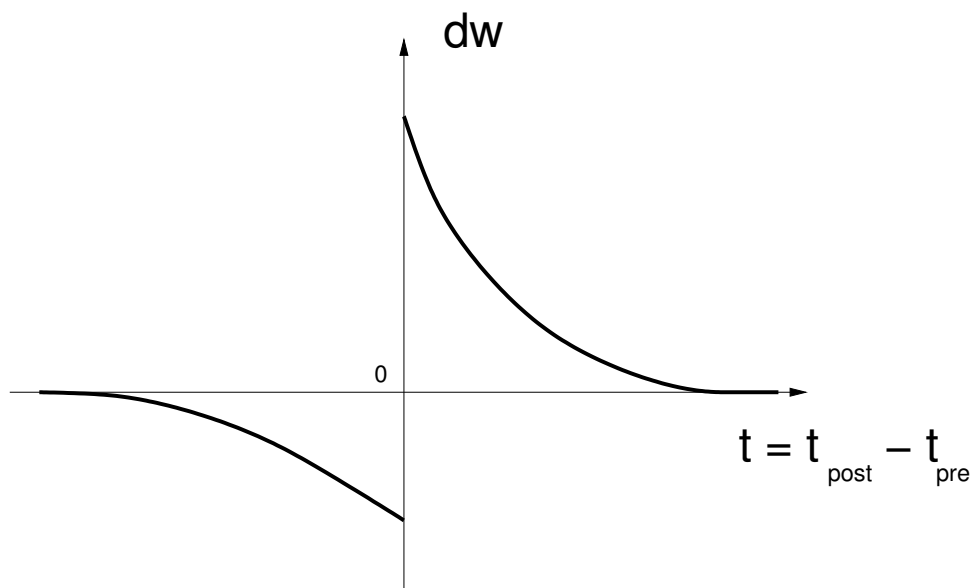


FIGURE 3.14 – La règle STDP définit la modification de poids dw subie par une synapse, en fonction de l'écart de temps qui sépare le potentiel d'action post-synaptique du potentiel pré-synaptique.

L'utilisation d'une règle STDP a été récemment utilisée pour étendre le modèle proposé dans [Dauce *et al.*, 1998], que nous venons de considérer. Les auteurs, dans cette extension, utilisent des neurones *integrate and fire* que nous avons évoqués au paragraphe 3.4.4, et leur imposent une règle STDP pour stabiliser la dynamique, ou une règle anti-STDP pour retourner vers un état caotique, selon un critère de renforcement positif ou négatif [Soula *et al.*, 2005; Henry *et al.*, 2007].

Ce type de règles, quand il est appliqué au sein d'une population de neurones de type Hodgkin-Huxley simplifiés (cf. paragraphe 3.4.1), conduit à la constitution de motifs spatio-temporels, ce qu'Izhikevich nomme *polychronie* [Izhikevich, 2006]. Les activations polychrones peuvent alors servir de base à la construction d'une machine à états liquides [Paugam-Moisy *et al.*, 2007].

32. *Spike-time-dependent Plasticity*.

33. Il s'agit de la règle BCM (Bienenstock-Cooper-Munroe), voir [Gerstner and Kistler, 2002].

3.7.4 Apprentissage par renforcement

Les travaux de Soula et collègues que nous avons mentionnés [Soula *et al.*, 2005; Henry *et al.*, 2007] sont un pas vers l'inclusion de l'*apprentissage par renforcement* pour paramétrer un système dynamique, en l'occurrence un robot Khepera devant éviter des obstacles à partir d'une image caméra. L'idée est de stabiliser le système en cas de renforcement positif, en renforçant les poids entre neurones actifs, et de déstabiliser le réseau en cas de renforcement négatif, en réduisant les poids entre les neurones du motif qui provoquent ce renforcement. Nous avons pour notre part proposé une démarche analogue [Ménard and Frezza-Buet, 2005] dans le modèle *bijama*, pour lequel la prédiction d'absence de récompense déstabilise les activités du réseau. Ces travaux restent à approfondir, ce qu'a commencé Thomas Legrand lors de son stage de Master [Legrand, 2006].

A notre connaissance, il n'existe pas aujourd'hui de modèle qui instancie, dans le cas des systèmes dynamiques, un apprentissage par renforcement aussi élaboré que les méthodes classiques de ce domaine [Sutton and Barto, 1998]. Les tentatives d'intégration de récompense dans les modifications de la dynamique sont instantanées, et ont simplement pour effet de retenir les configurations favorables (stabilisation) et d'exclure les autres (déstabilisation).

3.7.5 Conclusion

Si nous revenons à la formulation de l'apprentissage sous la forme de l'équation 3.11, le fait de modifier θ peut conduire à des bifurcations de la dynamique du système. Dans le cas des champs de neurones dynamiques, les études menées par Amari et Taylor (voir paragraphe 3.6.2) montrent la sensibilité de la dynamique au profil des poids latéraux, ce qui rend délicat leur apprentissage car la propriété de réaliser des bulles d'activité peut être perdue, au profit d'un champ complètement saturé par exemple, ou d'instabilités de Turing. Or de telles modifications de poids latéraux sont décrites en biologie comme étant une caractéristique de la plasticité corticale [Burnod, 1989], puisque les colonnes corticales³⁴ se couplent ou se découplent en modifiant leurs influences latérales.

Autant, pour un système dynamique donné, l'existence de bassins d'attraction vers des points fixes ou des cycles semble garantir de bonnes propriétés de débruitage ou de restitution de cohérence, autant il reste aujourd'hui difficile de trouver une procédure qui ajuste un système dynamique pour un problème donné. Face à cette difficulté, qui est au cœur de notre problématique de recherche, nous avons choisi un compromis en autorisant une injection de connaissance dans les couplages de différents champs dynamiques, mais en laissant, au sein de ces champs, des processus d'auto-organisation dédiés aux traitements requis pour la tâche, ce que nous détaillerons ceci au paragraphe 4.5. Il est heureux toutefois de constater que d'autres auteurs ont des approches très différentes de cette problématique, tant il paraît nécessaire, dans l'état actuelle des connaissances, d'explorer en largeur la notion d'apprentissage dans les systèmes dynamiques connexionnistes. Si, pour ce faire, l'on souhaite s'inspirer de la dynamique des systèmes neuronaux biologiques, la référence aux *ganglions de la base* est inévitable, ce que nous n'avons pas encore abordé dans nos recherches. Nous discutons ce point aux paragraphes 5.2.3 et ??.

34. que modélisent les points du champ de neurones.

3.8 Simulation des systèmes dynamiques

Avant de conclure ce chapitre consacré aux systèmes dynamiques, nous nous proposons dans ce paragraphe d'aborder la question de leur simulation. Il y a peu d'outils standards pour simuler ces systèmes autrement que par des scripts `MatLab` ou des développements ad hoc. Pour ce qui concerne les champs de neurones dynamiques, les auteurs mentionnent l'utilisation de méthodes de type Runge-Kutta, tout comme dans le cas des *Cellular Neural Networks*, mais des outils plus génériques existent pour dépasser le stade de la simple résolution numérique ad hoc.

Le logiciel `XPP-aut` permet par exemple de simuler des systèmes dynamiques régis par une équation différentielle [XPP-auth, 2008], procédant aussi par intégration numérique de l'équation. Une de ses originalités est qu'il possède un module permettant de trouver numériquement les points de stabilité³⁵, de linéariser le système en un point, et surtout de détecter les bifurcations en lançant une succession d'analyses de stabilité pour des valeurs différentes des paramètres du système.

On trouve également, dans le domaine des modélisations de propagation de potentiels de membranes régis par des équations dérivées du modèle de Hodgkin et Huxley, des simulateurs permettant de faciliter la mise en œuvre de gros systèmes. Ces équations régissent un fragment de potentiel de membrane, et la simulation d'un axone requiert de coupler plusieurs fragments mis bout-à-bout (cf. paragraphe 3.4.1). C'est ce qu'on appelle des *modèles compartimentaux*, et simuler un arbre dendritique revient à simuler un grand nombre d'équations différentielles (plusieurs par compartiment), en couplant certaines avec d'autres, de sorte que le graphe de couplage reproduise l'arbre. Des logiciels comme `NEURON`, `GENESIS` ou `BRIAN` permettent de simuler de tels systèmes, en se basant sur un langage de programmation dont les objets sont les compartiments, voire les canaux ioniques [NEURON, 2008; GENESIS, 2008]. Ces logiciels réalisent l'intégration des équations différentielles par des méthodes de type Euler et Runge-Kutta. Ces outils sont très utiles aux modélisateurs de systèmes biologiques du type de ceux décrits dans le paragraphe 3.4. Une revue de ce type de simulateurs est présentée dans [Brette et al., 2007].

À un niveau de modélisation plus grossier, on trouve le simulateur `NEST` [NEST, 2008], qui modélise des neurones impulsionnels du type de ceux vus au paragraphe 3.4.4, en autorisant la simulation de dizaines de milliers de neurones, sur des clusters, ainsi que l'implémentation de règles d'apprentissage de type STDP. `NCS` est un simulateur aux ambitions équivalentes [NCS, 2008], ainsi que `SPLIT` qui a été porté sur les supercalculateurs Blue Gene d'IBM [Djurfeldt et al., 2008].

Mentionnons également `CSIM` qui est moins dédié à la simulation de réseaux de neurones à proprement parler, mais qui permet de simuler des systèmes événementiels asynchrones organisés sous forme de hiérarchies [CSIM, 2008].

Enfin, dans le cadre des réseaux impulsionnels basés sur les modèles que nous avons présentés au paragraphe 3.4.4, ont été développés des *simulateurs événementiels*. Le principe est de considérer que le système, entre deux émissions d'un potentiel d'action dans le réseau, évolue de façon continue, les discontinuités étant des *événements*. Ces simulateurs doivent donc calculer la date du prochain événement, et il en existe des versions parallèles [Rochel and Martinez, 2003].

Pour ce qui nous concerne, nous implémentons les champs de neurones par une procédure d'évaluation asynchrone des unités (cf. paragraphe 3.3.2), ce qui ne rentre pas dans le cas des méthodes précédemment citées. Les réseaux de Hopfield, d'ailleurs, sont à situer dans cette catégorie également. Se pose alors le problème de l'énorme quantité de connexions à stocker,

35. Recherche basée sur une méthode de Newton.

surtout lorsque les poids latéraux sont sujets à modification et que le partage des valeurs de poids entre les unités est impossible. Il nous a donc été nécessaire, en travaillant avec Stéphane Vialle, de Supélec à Metz, et Jens Gustedt, du Loria à Nancy, de concevoir un simulateur de systèmes dynamiques à grain fin, que nous détaillerons aux paragraphes 4.4.3 et 9.7, et qui a motivé le projet *InterCell* présenté au paragraphe 8.1.2. Une collaboration avec Nicolas Fressengeas du LMOPS à Metz a permis de construire, au dessus de ce simulateur, un système de résolution d'équations aux dérivées partielles présenté au paragraphe 6.1, avec lequel on pourrait³⁶ simuler des systèmes dynamiques du type de ceux que traite *XPP-aut*.

3.9 Questions pour notre problématique

Nous concluons ce chapitre concernant les systèmes dynamiques par la précision de deux questions relatives à notre problématique. La première est ce que nous attendons des systèmes dynamiques pour notre projet de recherche, car les potentialités de ces systèmes sont, comme nous l'avons vu, très larges. La deuxième question est celle de la pertinence du formalisme impulsionnel, car nous avons fait le choix de rester dans un cadre fréquentiel, ce qui signifie que nos unités s'échangent des valeurs flottantes, plutôt que des impulsions.

3.9.1 Que fait la dynamique ?

Nous avons opposé dans ce chapitre deux interprétations de la dynamique d'un système. La première consiste à considérer cette dynamique comme un déroulement temporel (cf. paragraphe 3.5), alors que la seconde consiste à la considérer comme l'itération d'un processus de résolution de contraintes (cf. paragraphe 3.6) que nous avons qualifié de relaxation. Dans le cas des simulations de champs neuronaux dynamiques avec délai, mais aussi dans celui des *Cellular Neural Networks*, la résolution via des méthodes de type Runge-Kutta d'une équation aux dérivées ordinaires en temps fait que ces champs de neurones ont un paramètre temporel qui correspond au temps qui s'écoule, dans le système biologique modélisé par exemple. Dans les approches basées sur la relaxation, dont on attend des propriétés comme celles illustrées sur la figure 3.13, il n'y a aucune raison de mesurer ce temps en secondes.

Qu'en est-il alors de la prise en compte du temps par des approches de relaxation ? Ces deux temps se confondent-ils ? Il s'agit de questions que nous n'avons pas encore discutées dans nos modèles, mais nous avancerons ici les points suivants. On peut dans un premier temps considérer que la relaxation est beaucoup plus rapide que l'évolution temporelle, et que la dynamique du système consiste à modifier un état d'équilibre en fonction d'un paramètre temporel. Cette approche est toutefois assez gênante, car gourmande en temps de calcul. On peut deuxièmement considérer que ces deux temps se confondent, le système dynamique étant en perpétuelle relaxation vers un point d'équilibre remis en cause à chaque instant. Ce dernier point de vue nous apparaît bien plus prometteur, en terme de capacité de calcul que l'on peut en attendre, et aborder ce point par la réalisation d'un modèle fait partie des développements futurs de notre projet de recherche, ce que nous rediscuterons au paragraphe 5.2.

3.9.2 Pertinence du modèle impulsionnel

Nous souhaitons dans ce paragraphe justifier notre choix de garder des modèles basés sur un codage fréquentiel, alors que les dernières tendances en modélisation biologique mettent l'accent

36. Nous n'avons pas à l'heure actuelle réalisé ces tests.

sur les modèles impulsionnels. La première des raisons est que notre recherche n'est pas une entreprise de modélisation biologique, comme nous l'avons précisé au paragraphe 2.2, et nous n'avons en conséquence pas obligation à se rapprocher de la façon dont les neurones échangent effectivement de l'information, tant que nous pensons pouvoir capturer la logique de ces échanges par un modèle fréquentiel.

D'ailleurs, l'apport des approches impulsionnelles pour la définition d'un nouveau type de calcul n'est pas si évident, et ces approches n'ont en général pas cet objectif informatique et se focalisent sur la modélisation d'observations biologiques. Mentionnons quelques exceptions, comme les détecteurs de synchronisations utilisés par Hopfield [Hopfield and Brody, 2000; Hopfield and Brody, 2001] pour réaliser la détection de séquences de phonèmes robuste à la distorsion temporelle, les travaux sur la modélisation de repérage de sources auditives par la chouette [van Hemmen and Kempter, 1998], basés eux-aussi sur des propriétés de synchronisation, ou enfin la notion de *codage par rang* introduite par Simon Thorpe [Thorpe and Gautrais, 1998], dont l'équation est très contestable d'un point de vue biologique.

En conséquence de quoi, notre position est de ne pas s'imposer la complexité d'un modèle impulsionnel tant que nous n'en ressentons pas l'utilité informatique. Il n'empêche que les difficultés que nous rencontrons dans la maîtrise des champs de neurones de nos modèles pourraient laisser penser que justement, le besoin se fait sentir, mais à l'heure actuelle, nous ne voyons pas en quoi le modèle impulsionnel peut nous faire avancer sur ce point, mais gardons l'œil ouvert...

Chapitre 4

Le calcul cortical

Il est aujourd'hui devenu classique de mentionner la quantité prodigieuse de neurones que nous possédons dans nos cerveaux, de même que les longueurs faramineuses de câbles qui les relient, alors qu'à y bien réfléchir, ces chiffres n'ont rien d'impressionnant en soi, tant il est clair qu'agglomérer des cellules petites dans le volume du crâne conduit nécessairement à des chiffres astronomiques, ces cellules étant microscopiques. On pourrait ressentir tout aussi bien le grisement des grands nombres en comptant les cellules du gros orteil humain, et nous ne doutons pas qu'il y siège une vascularisation complexe, des cellules pouvant se régénérer en cas de blessure, sachant produire avec harmonie la kératine qui constitue l'ongle imposant dont cet appendice est affublé. Pourquoi alors cet émerveillement particulier pour la nature du cerveau, alors qu'il partage avec bien d'autres parties du corps la propriété d'être une organisation gigantesque et harmonieuse de cellules minuscules ?

En ce qui nous concerne, cet émerveillement provient du fait que le cerveau, et plus généralement le système nerveux, est un organe dédié à traiter l'information, et est de fait le siège de la conscience. Même s'il existe bien entendu des échanges d'informations au sein du gros orteil, assurant par exemple l'homéostasie, il reste très particulier au système nerveux d'être une circuiterie électrique de contrôle. Ce qui est particulièrement « *beau* » dans le cerveau n'est pas la quantité de cellules qu'il renferme, mais que cette foule arrive à s'organiser pour produire les comportements, le langage et la conscience. C'est bien parce que la tâche d'avoir à organiser autant de ressources en un tout cohérent nous paraît à nous, informaticien, difficile, que ces chiffres nous désarçonnent, d'autant plus lorsque l'on sait que tout part d'une seule cellule-œuf indifférenciée hébergeant environ trois milliards de paires de nucléotides. Dit autrement, la description d'un humain, où du moins du germe qui le produit, tient sur une simple clé USB de 4GB, que l'on peut s'offrir pour une vingtaine d'euros. Nous dirons en conséquence que si la nature fait quelque chose, c'est que c'est forcément « *simple* », au sens ici où il suffit de laisser se dérouler les lois de la physique sur la cellule-œuf pour fabriquer un être humain. Tout ce que l'on saura comprendre de cette simplicité aura des répercussions technologiques mais aussi philosophiques, et proposer des architectures de calcul inspirées du cerveau, comme nous le faisons, est une démarche d'investigation visant à dénicher la simplicité du système nerveux pour la restituer sur des systèmes de traitement de l'information manufacturés.

Lorsque l'on se confronte à la richesse de la littérature décrivant les fonctionnalités des différentes parties du cerveau, on a la sensation que tout est nécessaire, et l'on serait tenté (quoique vite rebuté par la complexité de la tâche) de vouloir modéliser tout le cerveau. D'autre part, il existe des animaux comme les crocodiles, au comportement situé, à la motricité complexe (nage et reptation), qui ne doivent pas l'élégance de leur comportement à la présence d'un gros

cortex dans leur boîte crânienne, et l'on pourrait considérer qu'il vaut mieux commencer à étudier les centres sous-corticaux, ce qui est déjà très ambitieux, avant d'entreprendre une étude sur le cortex.

Face à ces questions, nous avons choisi de nous concentrer malgré tout sur le cortex, qui n'est certainement ni plus ni moins compliqué qu'une autre structure nerveuse, voire qu'un gros orteil, et ce choix est une *hypothèse de travail*. Il est motivé par le caractère générique de cette structure, très bien défendu d'un point de vue biologique dans [Binzegger *et al.*, 2005], qui est le reflet selon nous du type de simplicité que nous cherchons. Donnons donc, pour entrer en matière, quelques chiffres concernant le cortex, et ne boudons pas l'émerveillement que ces chiffres provoquent.

Vernon Mountcastle décrit le cortex comme une structure bidimensionnelle [Mountcastle, 1997], comparable à une feuille chiffonnée, comme on le voit sur la partie gauche de figure 4.1. L'épaisseur de cette feuille est de 3 à 4 millimètres, et sa surface fait environ 2600cm^2 , ce qui nous fait tout de même un peu plus d'une feuille A3 pour chaque hémisphère. Cette feuille est constituée par la juxtapositions de filaments perpendiculaires à sa surface, comme le sont les poils d'un tapis, ce qu'illustre la partie droite de la figure 4.1. Ces filaments sont des circuits neuronaux constitués d'une dizaine de neurones [Jones, 2000], chacun d'eux formant ce que l'on appelle une *microcolonne corticale*. Le diamètre d'un de ces filaments est de l'ordre de $80\mu\text{m}$ ³⁷, ce qui correspond à une densité de microcolonnes, sur la feuille corticale, qui est d'environ $150\mu\text{col}/\text{mm}^2$. L'ensemble du cortex compte à lui seul approximativement 28×10^9 neurones, autant de cellules gliales, et une quantité de synapse estimée à 10^{12} [Mountcastle, 1997].

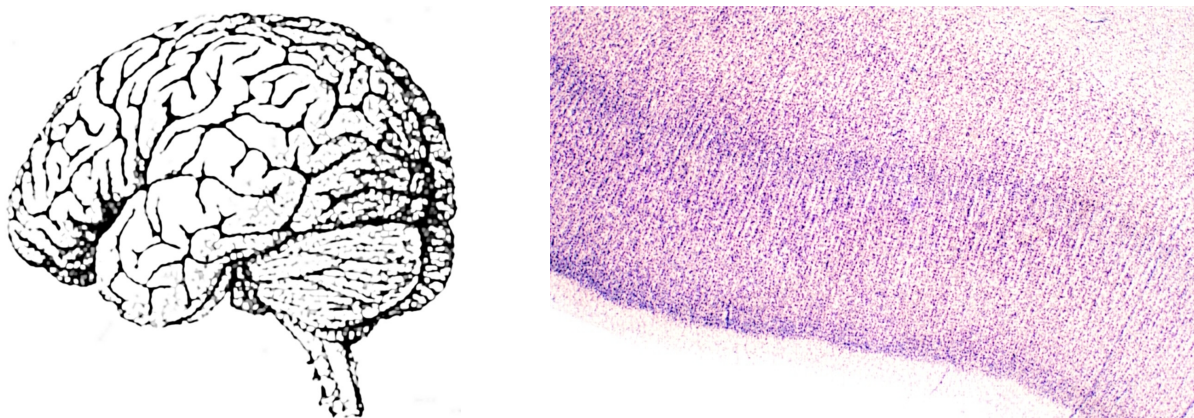


FIGURE 4.1 – À gauche, vue du cerveau humain, où le cortex apparaît, du fait de ses nombreuses circonvolutions, comme une feuille chiffonnée, d'après [Hubel, 1994]. À droite, coupe au travers de la surface corticale, laissant apparaître qu'elle est la juxtaposition de filaments, les microcolonnes, d'après [Jones, 2000].

Mountcastle décrit un niveau d'organisation un peu plus intégré, considérant comme unité constitutive la *minicolonne corticale*, qui regroupe une dizaine de microcolonnes. En un point de la surface corticale, on peut considérer que les colonnes corticales reçoivent des entrées similaires, provenant des mêmes sources, et elles sont liées par des connexions courte portée de sorte à former des groupes de colonnes que l'on peut considérer comme des modules de traitement d'une information. Ces groupes³⁸ sont appelés *maxicolonne corticale*, ou tout simplement *colonne corticale*. On retrouve également ces concepts dans [Burnod, 1989].

37. Il s'agit, pour être précis, de la période spatiale suivant laquelle ce circuit se répète.

38. Qu'il soit clair ici que c'est une notion locale, et donc « *glissante* » d'une minicolonne à sa voisine.

Dans le cheminement de nos recherches, nous tentons de donner un sens computationnel à la régularité de l'organisation corticale. C'est l'idée que nous développons le présent chapitre, en commençant par cerner ce que pourrait être un module cortical, impliqué dans une architecture corticale, présentée ensuite. Troisièmement, nous donnerons notre interprétation fonctionnelle du tissu cortical, pour enfin présenter le modèle `bijama`, qui est la dernière production de nos recherches dans ce domaine.

4.1 Module cortical

La structure corticale est similaire pour toutes les espèces, ce qui diffère de l'une à l'autre est l'expansion du cortex [Mountcastle, 1997]. On a d'ailleurs localisé récemment un gène impliqué dans le développement du cortex dont l'évolution a été rapide depuis notre ancêtre commun avec le chimpanzé [Pollard *et al.*, 2006]. Ceci corrobore l'idée qu'il est pertinent, au regard des critères optimisés par l'évolution dont nous sommes le fruit, d'étendre la structure corticale, sans qu'il soit nécessaire d'en changer la nature.

« Columns only vary from 300 to 600 μm in diameter, even between species whose brains differ in volume by a factor 10^3 . Cortical expansion in evolution is achieved by expanding cortical surface area, with little change in thickness. This expansion is generated by an increase in the number of cortical columns, not in individual column size » .

Vernon Mountcastle [Mountcastle, 1997]

On constate, depuis les travaux de Brodmann au début du XX^e siècle, que l'on peut identifier sur la surface corticale des régions, dont les contours sont toutefois mal définis et très variables d'un individu à l'autre, qui prennent en charge des composantes différentes du comportement (cf. figure 4.2). Des méthodes récentes, alliant observations à l'IRM fonctionnelle et une technique numérique de dépliement des cartes corticales permettent de comparer cette organisation entre cartes chez le singe et chez l'Homme [Orban *et al.*, 2004]. Les fonctions des différentes régions sont déduites, entre autres techniques, d'observations à l'IRM fonctionnelle, ou par l'étude de l'effet de lésions. Le cerveau étant un système très connecté, il est difficile de conclure quant au rôle d'une zone en observant les effets de la lésion de cette zone, ce qui est discuté par exemple dans [Friston, 2002]. Il semble tout de même qu'il y ait une certaine modularité au sein de la surface corticale, en ce sens que les traitements de différents types d'information s'y trouvent localisés en différents endroits. L'articulation de cette modularité dans le système nerveux, même si elle n'est pas encore complètement comprise par les biologistes, fait partie des propriétés du cortex particulièrement intéressantes pour les sciences des systèmes, et donc pour notre recherche, comme nous l'avons précisé au paragraphe 1.3.3.

Nous étudions donc dans ce paragraphe un module cortical, une carte, c'est-à-dire une portion de sa surface, que nous détachons du reste parce qu'elle nous semble réaliser une parmi les différentes fonctions opérées sur la surface corticale. Il s'agit bien entendu d'une vue de l'esprit, qui devient réalité dans notre travail de modélisation puisque nous considérons effectivement, comme structure informatique, la carte corticale, dont notre système possède plusieurs instances que nous relient. Cette modélisation ne doit toutefois pas faire oublier que sur le cortex, il s'agit plutôt d'un continuum de fonctions, et qu'y délimiter des régions fonctionnelles est toujours une entreprise empreinte de subjectivité.

Si l'on admet alors que le cortex est constitué de plusieurs cartes d'une part, et que ces cartes sont génériques d'autre part, il convient de se demander ce qu'est la nature commune au calcul

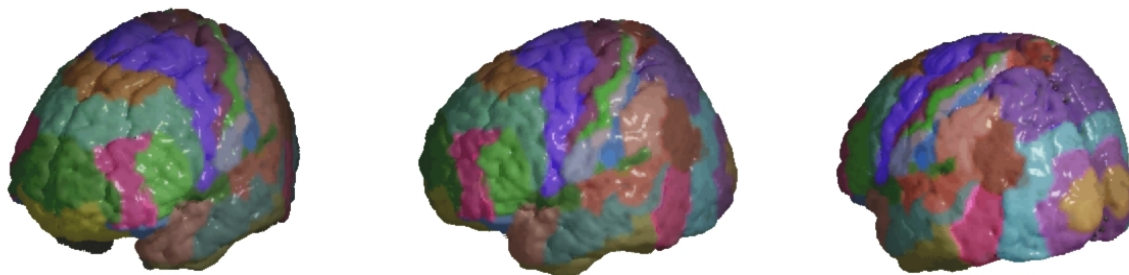


FIGURE 4.2 – La surface corticale est pavée de régions aux fonctionnalités différentes, comme l’a proposé Brodmann. Cette vue d’artiste provient du site <http://lcni.uoregon.edu/~mark/>.

effectué au sein de ces cartes, calcul qui peut se décliner en diverses modalités. C’est ce point qu’aborde ce paragraphe.

Avant d’entrer en matière, soulignons que la question de voir, dans la diversité des modalités traitées au sein des grandes structures du système nerveux, une fonction commune, mais appliquée à des données différentes, a été soulevée dans la littérature. Par exemple, Cummings propose de considérer les circuits reliant le cortex préfrontal, les ganglions de la base et le thalamus comme une même fonction, gérant des données différentes selon les endroits considérés [Cummings, 1995]. Plus récemment, Doya a formulé une description fonctionnelle des grands centres du cerveau³⁹, que sont le cortex, le cervelet, les ganglions de la base et le thalamus [Doya, 1999].

« [...] anatomical features, which are preserved throughout each of these structures, suggest that the cerebellum, the basal ganglia, and the cerebral cortex are each specialized for a certain kind of computation. However, the question about their unique « purposes » of computation, such as motor control, sensory acquisition, or cognitive processing, is ill-posed because all of these functions depend on each other under a normal behavioral context. [...] The approach [...] is to ask what are the unique « methods » of computation, namely, the learning algorithms, for which the cerebellum, the basal ganglia and the cerebral cortex are specialized. » .

Kenji Doya [Doya, 1999]

4.1.1 Un pavage bidimensionnel d’unités

Nous avons mentionné que le cortex est un pavage bidimensionnel de minicolonnes, et se pose en biologie la question soulevée dans [Jones, 2000], de savoir si l’on peut conclure, d’après les observations du cortex, à l’existence d’un tissu générique constituant toutes les aires corticales, ces dernières ayant pourtant des fonctions différentes.

« The jury is still out on whether anatomical microcolumns are fundamental units of organization in all cortical areas of all species and, until we have more comprehensive data on the intricacies of intracortical connectivity, it may be very premature to view anatomical microcolumns as indicative of a fine-grain functional modularity of cortex, although many of us find this to be an attractive hypothesis » .

Edward J. Jones [Jones, 2000]

39. L’article ne traite pas de l’hippocampe toutefois.

Il semble qu'en biologie, la prudence soit encore de mise sur cette question, et que les connaissances actuelles ne permettent pas de franchir le pas de l'affirmation de l'existence, dans notre cortex, d'une sorte « *d'automate cellulaire neuronal* » homogène en structure, mais pouvant traiter n'importe quelle modalité comportementale. Comme le dit Jones, il s'agit toutefois d'une idée soutenue par plusieurs biologistes, comme Miller et collègues qui montrent l'analogie de structure entre le cortex somatique et le cortex visuel du chat [Miller *et al.*, 2001], ou Ballard qui soutient explicitement cette idée [Ballard, 1986].

Notre projet de recherche, tel que nous l'avons énoncé au paragraphe 2.2.2, consiste à franchir ce pas.

Ce faisant, dans la mesure où notre problématique est informatique, nous ne sommes pas redevable vis-à-vis de la communauté scientifique de la plausibilité biologique de cette approche, mais nous devons en revanche faire la preuve que postuler l'existence d'un calcul cortical générique permet de faire progresser l'informatique. Les mentions au cortex faites dans ce chapitre, et plus généralement dans ce document, sont empreintes de cette exigence.

Si l'on part du postulat que la microcolonne ou la minicolonne corticale est le composant élémentaire de la surface corticale, il est utile de rappeler l'organisation des connexions que reçoit et envoie ce composant. C'est là qu'intervient la structure « en couches » de la feuille corticale, comme illustré sur la figure 4.3, établie à partir des descriptions données dans [Burnod, 1989, pages 67–69]. Il semble alors, d'après cette description, que la minicolonne corticale ait à gérer des flux d'information, en provenance d'origines diverses, et à renvoyer de l'information vers divers centres également. Les connexions sont regroupées, dans les couches, en fonction de leur provenance, ce qui a une conséquence sur la modélisation que nous en faisons. En effet, dans nos modèles, depuis nos travaux de thèse [Frezza-Buet, 1999; Frezza-Buet and Alexandre, 2002], nous considérons que les unités d'un module cortical, qui sont le pendant des minicolonnes, ont des *connexions typées*, au sens informatique du terme, puisqu'elles se conçoivent comme appartenant à un ensemble. Nous reviendrons sur cette question au paragraphe 4.4.3.

Terminons notre description d'un module cortical comme un pavage de minicolonne en rappelant ce que nous avons dit au début du paragraphe 4.1 en nous appuyant sur la remarque de Mountcastle. Nous sommes l'objet d'une évolution qui s'est effectuée en agrandissant le cortex par un ajout de surface, et non par un changement de la nature des minicolonnes. En considérant les cartes corticales comme des modules, on peut interpréter ceci comme le fait que l'évolution rajoute des modules corticaux pour augmenter les aptitudes du cerveau, sans que les modules que possède un animal à gros cortex ne soient fondamentalement différents de ceux d'un animal à petit cortex. C'est ce point qui a inspiré la notion de *ressources de calcul extensives* que nous avons présenté au paragraphe 2.2.2 comme au cœur de notre projet de recherche.

4.1.2 Quelle fonction pour un module ?

Comme discuté dans [Friston, 2002], une conception trop localiste des modules, telle qu'elle peut être suggérée par les études lésionnelles par exemple, est à nuancer. Dire que les aires visuelles gèrent la vision, les aires somesthésiques le corps, est certainement réducteur, et ce n'est pas sous cet angle-là que nous aborderons la fonction du module cortical. Au contraire, à l'instar de Doya [Doya, 1999], nous proposons une fonction générale, pour un module. Nous ne pouvons affirmer aujourd'hui que c'est cette fonction-là qui est réalisée dans le cortex, mais c'est celle que nous implémentons dans les modules informatiques que nous proposons.

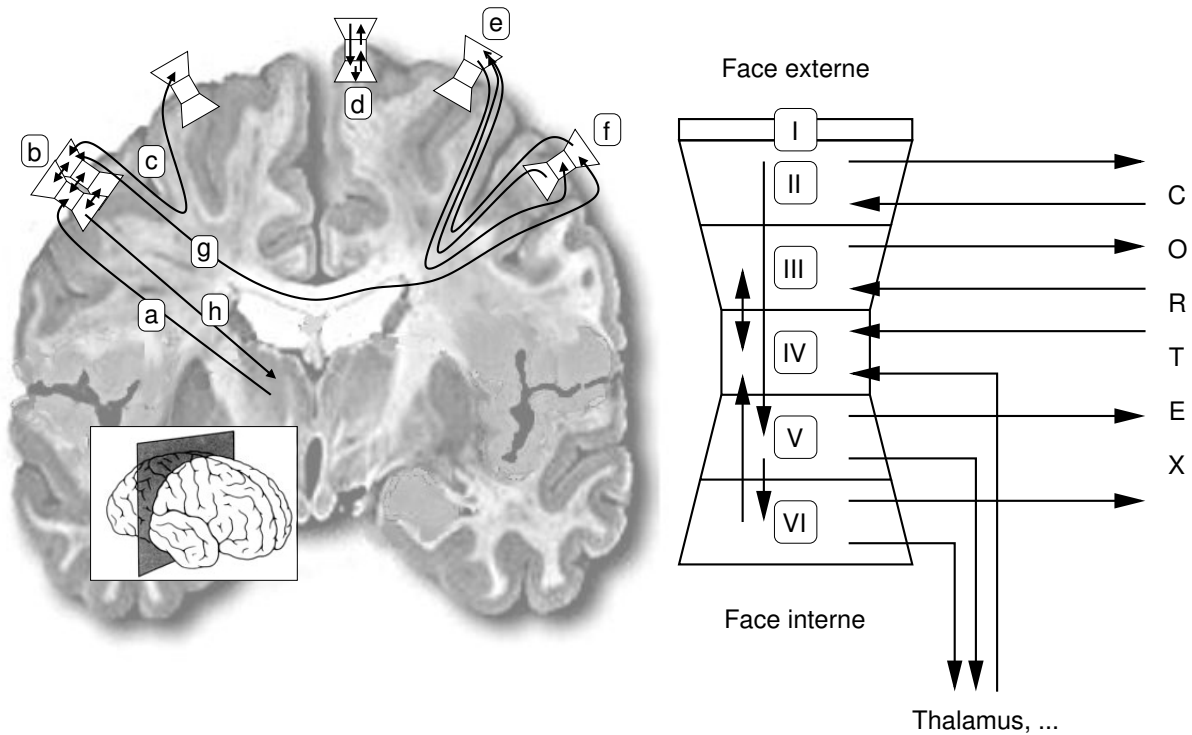


FIGURE 4.3 – Schéma d'une minicolonne corticale et de ses connexions. À droite, une minicolonne est représentée, avec ces six couches. La couche II projette vers des aires corticales éloignées, de façon ipsilatérale (c) ou contralatérale (g). La couche III assure la connexion aux zones du cortex adjacentes sur la surface. La couche IV, reçoit des informations provenant du thalamus (a) quand elle appartient à une aire sensorielle, ou alors du cortex, sous forme de *feedback*, quand il s'agit d'une aire qui n'est pas directement sensorielle (f). Les couches V et VI projettent vers les centres sous-corticaux (h) ou alimentent en *feedforward* d'autres zones corticales (f). Les colonnes adjacentes sont couplées au niveau de toutes les couches (b), et il existe des relations « verticales » au sein de la colonne (d). Enfin, il existe aussi des connexions des couches supérieures qui, contrairement au cas (e)-(f), sont réciproques et symétriques, formant un réseau récurrent (c).

Pour nous qui portons un regard informaticien sur le cerveau, il apparaît que son rôle est de trier et *sérialiser* l'information sous la forme de la production d'un comportement, et nous renvoyons le lecteur au paragraphe 1.3.2 pour un développement de ce point de vue. Ainsi, nous pensons que le cortex doit, en chaque point de sa surface, réduire l'information qu'il reçoit. Or si chaque module produit une réduction, les premiers modules qui reçoivent l'information ont une activité lourde en conséquence, puisqu'ils vont filtrer ce qui est transmis aux autres. C'est pourquoi il nous paraît plus plausible de parler de l'action d'un module comme d'une *restitution de cohérence* concertée avec les autres modules, ce que met en avant le modèle *bijama* que nous présentons au paragraphe 4.5. Les systèmes dynamiques que nous avons mentionnés à ce propos au paragraphe 3.6 et en particulier les champs de neurones dynamiques (cf. page 57), montrent comment l'on peut réduire l'information sans être tributaire de l'arbitraire d'un seuil. Ainsi considérons nous le module comme un champ de neurones dont les unités, représentant les minicolonnes, s'appuient sur les connexions latérales pour filtrer les flux d'information dont elles ont la charge. L'activité d'une unité, qu'elle transmet en sortie aux autres zones qui en dépendent, est le fruit d'une relaxation, d'une *compétition* aux critères multiples. D'ailleurs même dans les aires visuelles, souvent décrites comme de simples filtres de Gabor bidimensionnels [Daugmann, 1985], les colonnes ont une sélectivité qui dépend des colonnes voisines, si bien qu'elles soient bien plus sélectives que ne le sont les ondelettes de Gabor. C'est ce qu'illustre le concept d'inhibition antiphase mis en avant par Miller [Troyer *et al.*, 1998].

Pour nous, la formidable capacité du cortex à transformer un flux d'information parallèle en un comportement sériel, articulé, voire chez l'homme en un raisonnement prédicatif situé, tient à la mise en musique des réductions d'information que proposent chacun de ces modules, et l'ajout d'un module revient à proposer à l'ensemble un argument de plus pour trouver collectivement la réduction la plus pertinente. Notre projet de recherche tend à valider cette idée en l'instanciant par un système dynamique à grain fin.

4.2 Architecture corticale

Nous appellerons *architecture corticale* la mise en connexion de modules, c'est-à-dire de cartes corticales pour prendre un terme de biologie. En biologie, cette connexion est principalement dirigée au moment de la formation du cerveau, ce qui est décrit pour le cortex dans [Mountcastle, 1997]. Ce développement cortical conduit aux grandes symétries qui donnent à la connectivité du cortex une structure d'hypercube [Burnod, 1989, page 157-161], du fait de six dépliements successifs de la structure épithéliale qui donne naissance au cortex. Un dépliement de plus, et le cortex serait deux fois plus gros [Mountcastle, 1997].

« Cortical areas are produced by cell divisions and they can be viewed as the end product of successive dichotomies of the cortical sheet. In each dichotomy, a cortical zone produces two new zones of about equal size. The main important aspect of cortical connectivity is that new connections are systematically formed between the two new cellular subsets and these connections are symmetrical about their border line : they link symmetrical columns in a parallel, systematic way. » .

Yves burnod [Burnod, 1989, page 157-161]

L'axe de symétrie de plus haut niveau est l'axe gauche/droit, séparant les deux hémisphères corticaux. Les fibres qui relient une région à sa région symétrique sont celles qui passent par le corps calleux, entre les deux hémisphères. Se retrouve ensuite une symétrie frontal/postérieur⁴⁰,

40. Burnod utilise le terme associatif pour le postérieur.

dont la « *pliure* » est le sillon central. Vient ensuite la symétrie ente parties médiales⁴¹ et les parties externes. On trouve ensuite la symétrie vision/audition, puis pariétal/temporal, et enfin, au niveau le plus bas, entre aires primaires, qui reçoivent directement l'entrée externe, et aires secondaires, juste « *un niveau au dessus* ». Cette dernière symétrie est celle qui oppose V1 à V2 (vision), S1 à S2 (toucher), M1 à M2 (moteur), A1 à A2 (audition).

Nous n'avons retenu de cette organisation, dans l'état actuel de la construction de nos modèles, que le fait que des faisceaux de connexions parallèles se croisent sur la surface corticale, réalisant ces six symétries. C'est ce qui a en effet inspiré la connectivité en *stripes* mise en avant dans le modèle *bi_jama* (cf. paragraphe 4.5.2). De plus, l'architecture corticale, sur un cortex mature, peut-être vue comme un a priori structurel pour analyser l'information, et c'est comme tel que nous le considérerons. *Ainsi, nous n'envisageons pas une approche où cette architecture serait calculée, par une approche évolutionniste par exemple. L'architecture est au contraire, dans nos travaux, une injection de connaissance de la part du concepteur du système cortical artificiel.*

4.2.1 Indices primaires et secondaires

Nous proposons dans ce paragraphe de focaliser à nouveau notre attention sur un module cortical, comme nous l'avons fait au paragraphe 4.1, mais en considérant le module non plus isolément mais effectivement au sein d'une architecture. On constate alors, que l'organisation de l'information sur la surface d'une carte corticale, que nous avons appelée un module dans notre approche informatique, peut se décrire suivant un système de coordonnées à quatre dimensions [Ballard, 1986]. Les deux premières coordonnées, que nous appellerons d'après Ballard *indices primaires*, désignent un point de la surface corticale. Plus précisément, ces coordonnées repèrent une région de minicolonnes, cette région étant définie par le fait que toutes les minicolonnes qui la composent reçoivent la même entrée. Nous avons appelé une région définie de la sorte une *maxicolonne corticale* au début de ce chapitre. Les coordonnées primaires n'ont donc pas vocation à repérer *une* minicolonne, mais plutôt une maxicolonne, et ces coordonnées sont donc un repérage grossier.

Au sein d'une maxicolonne, qui est donc une petite portion de surface, on peut repérer les minicolonnes par un autre système de coordonnées, bidimensionnel aussi, et ce système, plus fin que le premier, constitue ce que nous appellerons les *indices secondaires*. Le repérage d'une minicolonne s'effectue donc en précisant les indices primaires de la maxicolonne à laquelle elle appartient, puis en précisant les indices secondaires de cette colonne.

Ce repérage par deux échelles de granularité peut paraître inutile, et en effet il ne prend son sens que si l'on utilise, comme indice, autre chose que les coordonnées d'un repère orthonormé que l'on aurait posé sur la surface corticale. Ce que l'on considère pour le repérage, c'est une description fonctionnelle de l'information, et c'est là que se justifie l'usage de quatre dimensions. Prenons l'exemple d'une de nos deux aires V1⁴² pour illustrer ce point. V1 est connectée au corps genouillé latéral⁴³ (LGN⁴⁴) de façon redondante. Ainsi, pour une petite région de V1, on peut en première approximation considérer que toutes les minicolonnes de cette région reçoivent des informations des mêmes neurones de LGN, ce qui fait de cette petite région de V1 une maxicolonne, par définition. Or LGN expose des informations provenant de la rétine. Ainsi, à chaque neurone de LGN, on peut associer, pour des raisons *anatomiques*, deux valeurs, qui sont d'une part le degré d'excentricité par rapport à la fovéa de l'information traitée et d'autre

41. Cela correspond à l'importante « *surface de contact* » entre les deux hémisphères

42. Nous avons un V1 gauche, et un V1 droit.

43. Celui des deux qui est du même côté du cerveau

44. *lateral geniculate nucleus* en anglais/latin.

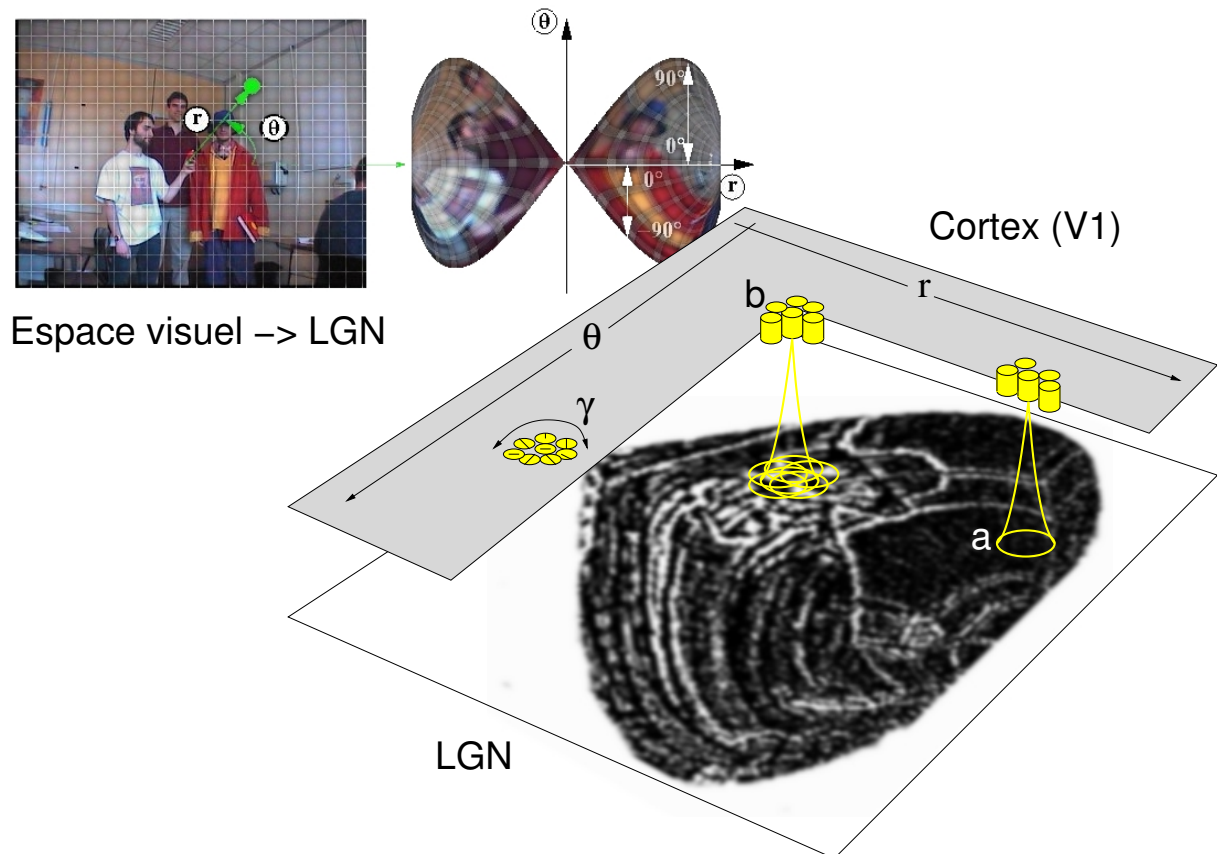


FIGURE 4.4 – Indices primaires et secondaires. L'espace visuel est projeté sur les corps genouillés latéraux selon le rayon et l'azimuth (la simulation est faite avec `lgn2v1`, voir paragraphe 9.6.4), comme illustré en haut de la figure. Les colonnes du cortex reçoivent une information des LGN via leur *champ récepteur* (a). Dans une maxicolumn (b), on peut en première approximation considérer que les champs récepteurs sont tous les mêmes. Les indices primaires sont r, θ , et les indices secondaires, sur ce schéma, sont l'orientation de contraste notée γ .

part l'azimut de cette information. Dit autrement, un neurone de LGN est relatif à une petite portion du champ visuel. C'est le repérage de cette portion, dans la scène visuelle, qui sert d'indice primaire, car la maxicolonne de V1 traite les neurones de LGN relatifs à cette portion. Sur la surface de V1, on repérera donc les maxicolonnes par leur degré d'excentricité et leur azimut, ce qui correspond à un étiquetage *fonctionnel* guidé par l'*anatomie* des connexions.

Les observations d'Hubel et Wiesel ont d'autre part montré qu'au sein d'une maxicolonne de V1, les minicolonnes repèrent des orientations spécifique des contrastes vus à l'endroit correspondant de la rétine, avec une sensibilité à un œil ou l'autre [Hubel, 1994]. La latéralité gauche/droit de l'œil et l'angle du contraste forment alors les indices secondaires de V1. Ainsi, une minicolonne de V1 se repère sur la surface par ce *référentiel fonctionnel* en précisant l'excentricité (primaire-1), l'azimut (primaire-2), l'œil (secondaire-1) et l'orientation (secondaire-2) de l'information traitée. Les rapports entre indices primaires et secondaires et l'architecture neuronale sont représentés sur la figure 4.4. On trouvera dans [Ballard, 1986] une table qui donne, pour différentes aires corticales, les indices primaires et secondaires (cf. table 4.1) qui en expriment l'organisation spatiale.

Ce qui est très intéressant selon nous d'un point de vue informatique, c'est que ce repérage fonctionnel est mis en bijection avec le repérage spatial (coordonnées métriques 2D) sur le tissu cortical. Cette propriété répond justement aux contraintes de répartition spatiale des calculs exprimées au paragraphe 2.1.3.

Indices primaires

La notion d'indices primaires correspond, comme nous l'avons vu et comme le rappelle la figure 4.4, à la projection d'une modalité sur la surface corticale, telle qu'elle résulte de l'*anatomie des projections axonales*. Il s'agit d'une mise en correspondance de deux modules neuronaux, par une fonction d'indices continue. Pour formaliser un peu, revenons à notre notion de *champ récepteur*, telle qu'elle a été introduite au paragraphe 3.3 qui diffère un peu de ce que recouvre ce terme en biologie. Considérons un module neuronal A , par exemple V1 sur la figure 4.4, relié à un autre module B , qui est LGN sur cette même figure. Considérons un étiquetage arbitraire des neurones qui, au sein des deux modules, assure que les étiquettes de neurones proches sont proches⁴⁵. Sur une surface, cet étiquetage peut être tout simplement les coordonnées (x, y) du neurone, par rapport à un repère 2D métrique. On notera $(x, y)_A$ les coordonnées d'un neurone dans le module A , et $(x, y)_B$ les coordonnées d'un neurone dans le module B . Soit φ la fonction qui, à l'étiquette $(x_a, y_a)_A$ d'une minicolonne de $a \in A$, associe l'étiquette $(x_b, y_b)_B = \varphi((x_a, y_a)_A)$ du neurone de $b \in B$ qui est au centre du champ récepteur de a . La fonction φ représente la connectivité entre les deux modules, et cette fonction représente par définition une connectivité *topographique* lorsque φ est continue. C'est le cas pour les connexions de nombreuses zones du cerveau, et en particulier pour le cortex, comme par exemple les aires, visuelles, auditives, somesthésiques et motrices [Mountcastle, 1997]. Sur la rétine, l'image se projette via une lentille, le cristallin, et la conservation de topologie entre la scène visuelle et les neurones est « *physique* ». De la rétine aux LGN d'une part, puis des LGN à V1 d'autre part, les connectivités sont topographiques, ce qui explique que l'on puisse utiliser l'excentricité et l'azimut, propriétés de la scène visuelle, comme système de coordonnées primaires dans V1.

45. Ce qui suppose une notion de voisinage sur l'espace des étiquettes.

Aire corticale	Indices primaires	Indices secondaires
Chez le macaque		
V1	Espace (rétinien)	Fréquence spatiale, couleur, orientation, dominance oculaire
V2	Espace (rétinien)	Mouvement, disparité
V4	Espace (rétinien)	Couleur
MT	Espace (rétinien)	Mouvement
FEF	Direction de la prochaine saccade oculaire	Direction de la précédente saccade oculaire
1, 3b	Surface du corps	Paramètres cutanés
3a, 2	Topologie de l'espace	Paramètres articulaires et muscles
Chez le chat		
17	Espace (rétinien)	Fréquence spatiale, orientation, dominance oculaire
18	Espace (rétinien)	Mouvement

TABLE 4.1 – Exemples d'indices primaires et secondaires pour quelques cartes corticales. D'après [Ballard, 1986].

Dans nos recherches, nous n'étudions pas l'établissement de connectivités topographiques, nous les construisons explicitement de la sorte. Nous renvoyons le lecteur intéressé par l'établissement auto-organisé d'une connectivité topographique entre deux modules neuronaux aux travaux de Christopher von der Malsburg [Willshaw and Malsburg, 1976; Zhu and von der Malsburg, 2004]. Nous souhaitons de plus citer les travaux de Bressloff et collègues [Bressloff *et al.*, 2002], qui illustrent le caractère topographique des étapes du traitement visuel par la nature des hallucinations du type de celles vécues sous LSD, ou lors des expériences de mort imminente. En effet, ces auteurs montrent qu'une image d'une spirale logarithmique en rotation, par exemple, produit, par la cascade des connectivités φ , des rayures « droites » sur la surface de V1 qui se déplacent comme des ondes planes. Selon ces auteurs, sous l'emprise de dysfonctionnement du cerveau, ce type d'ondes peuvent apparaître sur V1, guidés simplement par des oscillations sur la surface de V1 sans aucune régulation par un stimulus visuel. On comprend alors pourquoi les sujets ont l'impression de glisser dans un tunnel, puisque un vrai glissement produirait les mêmes activités sur V1.

Indices secondaires

Nous avons vu que la définition d'indices secondaires est présente au sein d'une maxicolonne. « *L'attribution* » d'indices secondaires est le fruit d'un processus d'auto-organisation, à l'instar de l'attribution des indices primaires, ce qui a été modélisé par Teuvo Kohonen [Kohonen, 1997, chap. 4]. Nous ne parlerons pas ici du célèbre algorithme de quantification vectorielle, les cartes auto-organisatrices, mais du modèle de maxicolonne qui a précédé son élaboration par Kohonen. Ce modèle décrit la maxicolonne comme un champ de neurones dynamique, analogue à ceux que nous avons présentés au paragraphe 3.6.2. Toutes les unités du champ reçoivent une même entrée, ce qui est cohérent avec la notion de maxicolonne. En revanche, les unités opèrent sur cette entrée un filtrage qui est propre à chacune d'elles. Ce filtrage est tel que l'unité répondra fortement à une configuration de l'entrée, que nous nommerons *prototype* de l'unité, et sa réponse décroît avec l'éloignement de la configuration de l'entrée par rapport au prototype. La forme de cette décroissance est ce que l'on appelle la « *courbe d'accord* » de l'unité. À un instant donné, pour l'entrée courante de la maxicolonne, les différents prototypes des unités les conduisent à produire une distribution de réponses sur la surface de la maxicolonne. Cette distribution joue le rôle d'entrée i dans un processus de *compétition* qui peut être décrit par l'équation 3.6. La distribution d'activité u résultante, qui a une forme de bulle si l'on modélise la maxicolonne par un champ dynamique (cf. figure 3.7), module l'apprentissage des prototypes des unités. Une unité active fait en sorte de mieux répondre à l'entrée courante qu'auparavant, ce qui revient à décaler son prototype vers l'entrée courante. Kohonen a montré qu'avec ce principe, on obtient une répartition de prototypes sur les unités de la maxicolonne qui est telle que les prototypes de deux unités proches sur la surface sont des prototypes similaires. C'est ce qu'on appelle la conservation de topologie, qui a été d'ailleurs exploitée dans les approches de quantification vectorielle (cf. paragraphe 6.2), et que l'on retrouve dans de nombreux modèles (cf. figure 4.5).

Ce qui se joue au sein de la maxicolonne, et qui a pour effet l'établissement d'indices secondaires, est la mise en adéquation des connexions de voisinage au sein de la maxicolonne avec la topologie des entrées successives que la maxicolonne a reçues. À l'instar d'autres auteurs [Douglas and Martin, 2007], nous interprétons la dynamique de la maxicolonne comme celle d'un champ neuronal dynamique, c'est-à-dire comme visant à concilier l'information reçue avec « *ce qu'elle devrait être* » au regard des apprentissages passés. Cette conciliation est une forme de *restitution de cohérence*, de même nature que celle que nous avons soutenue au paragraphe 3.6.2. Se pose toutefois le problème de l'acquisition, par apprentissage, de la cohérence que la relaxation

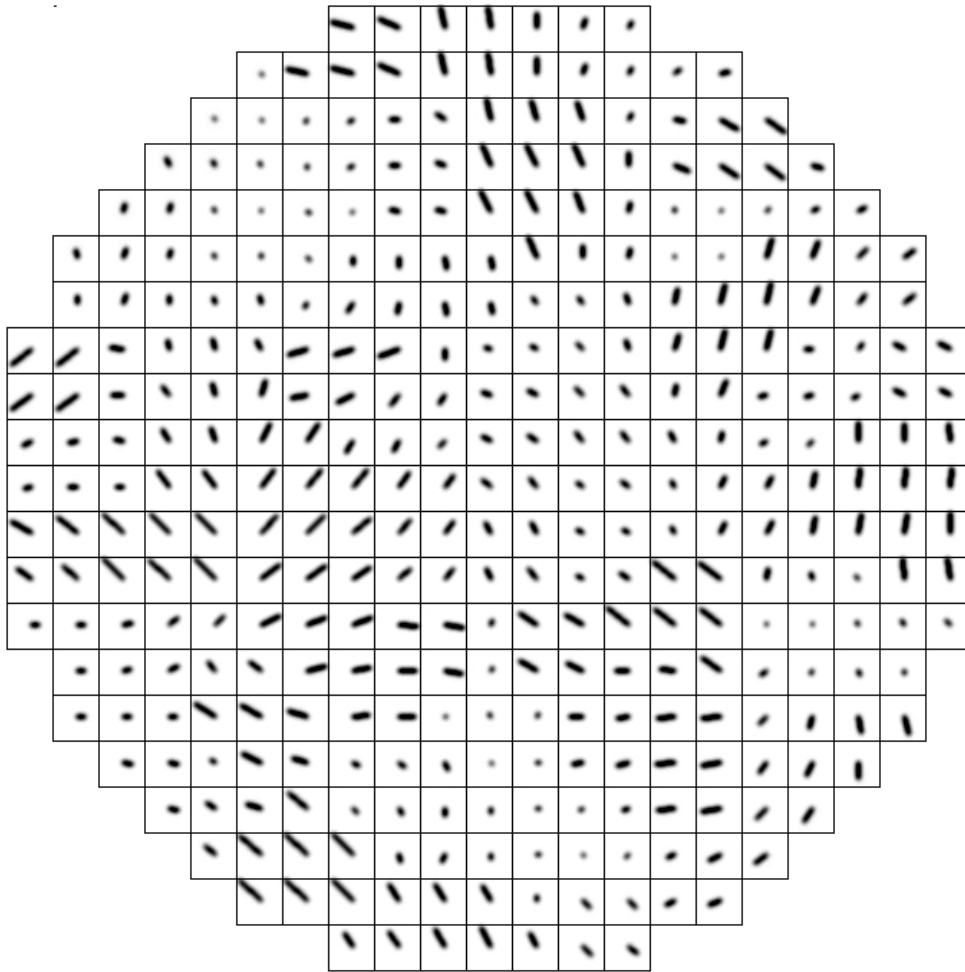


FIGURE 4.5 – Maxicolonne, modélisée par notre bibliothèque `bijama`, qui reçoit des vecteurs de déplacement 2D. À l'endroit de chacune des unités de la maxicolonne est dessinée l'orientation qui est le prototype de l'unité, après un traitement analogue au principe de Kohonen décrit dans le texte.

restituée à chaque instant.

Dans le cadre des champs neuronaux dynamiques, cette cohérence est le profil *on-center/off-surround* des connexions latérales, et il est imposé, forçant la relaxation à se stabiliser sur des activités isolées. L'apprentissage consiste alors à ajuster les prototypes, qui correspondent dans les modèles à l'activation de la couche IV de la colonne. C'est là que se situe le modèle de Kohonen, mais aussi d'autres approches plus récentes, se référant plus explicitement à l'analyse des systèmes dynamiques, et où des notions de rupture de symétrie lors de l'établissement des prototypes sont mises en avant [Bressloff, 2005; Lücke and Bouecke, 2005b].

On peut au contraire dans les modèles fixer les prototypes sur la surface, et apprendre leur corrélation spatiale. L'intérêt est alors d'utiliser la relaxation pour « régulariser » un profil d'activité bruité [Baraduc, 1999]. Dans ces travaux toutefois, la relaxation revient à effectuer une projection, en « *une passe* », ce qui s'éloigne de la dynamique complexe des champs neuronaux.

Nous avons tenté dans le passé, lors du stage de DEA de Yann-Éric Douvier [Douvier, 2003], d'autoriser à la fois l'apprentissage des liens latéraux (couplage entre minicolonnes) et des liens *feedforward* (définition des prototypes). Nous avons constaté que la dynamique des champs neuronaux était détruite par les apprentissages latéraux, et n'avons pu mener à bien cet apprentissage. Au prix d'une simplification des entrées (gaussiennes orientées par exemple) et d'une algorithmique parfois très ad hoc, d'autres auteurs ont pu réaliser cet apprentissage conjoint, comme notamment dans [Fellenz and Taylor, 2002], et surtout dans les dérivés du modèle RFLISSOM, rassemblés dans [Miikkulainen et al., 2005]. Ces travaux, qui sont bien plus portés que les nôtres sur une modélisation de données biologiques, se sont autorisés des simplifications algorithmiques⁴⁶ que nous nous interdisions, dans la mesure où notre approche, elle, se justifie par sa portée pour la science informatique.

À notre connaissance, les modèles de maxicoulonne corticale abordent la complexité de l'élaboration d'une répartition « *continue* » de prototypes, comme ceux de la figure 4.5, et considèrent le module cortical isolé. L'une des originalités de nos travaux avec Olivier Ménard sur *bijama* est de montrer que cette organisation en indices secondaires peut aussi être le reflet d'une cohérence plus générale, qui dépasse l'enceinte du module. Nous justifierons ce point au paragraphe 4.5.5.

Pour conclure ce paragraphe, nous proposons de considérer la notion d'indices secondaires comme le reflet d'un *codage tabulaire* de l'information. Ce qui s'organise au niveau de la constitution des indices secondaires, c'est la prise en charge non supervisée par chacune des minicolonnes d'une maxicoulonne de la représentation d'une configuration de l'information que, toutes, elles reçoivent. Ce codage tabulaire, proche de ce que l'on appelle en mathématiques la quantification vectorielle, s'oppose au neurone formel qui filtre l'information par une fonction monotone. Réaliser un apprentissage par un *codage monotone* a toutefois des avantages bien connus en apprentissage automatique, en particulier pour extrapoler ce qui est appris en dehors de la zone d'apprentissage. C'est cet argument qui pousse certains auteurs comme Emmanuel Guigon, avec qui nous avons collaboré (cf. paragraphe 8.1.4), à faire la promotion d'un codage monotone dans la prise en charge par le cortex des commandes motrices [Burnod et al., 1999; Baraduc et al., 2001; Baraduc and Guigon, 2002], justifiant ce choix par des expérimentations qui révèlent des capacités d'extrapolation des commandes. Ce point de vue va toutefois à l'encontre de l'uniformité du cortex, qui en ce qui concerne la vision montre clairement un codage tabulaire. D'ailleurs, Dana Ballard tranche la question dans le sens du codage tabulaire :

46. Absence de parallélisme, différentes étapes dans le déroulement de l'apprentissage, connectivité non constante, etc.

« *The brain may have both value [codage tabulaire] and variable [codage monotone] neurons or use combined strategies, but the gross organization of the cortex seems to exhibit value encoding* » .

Dana H. Ballard [Ballard, 1986]

Faut-il faire une exception pour les maxicolonnes du cortex moteur, qui lui, serait le siège d'un codage monotone, et pour lequel, par conséquent, la notion d'indices secondaires ne s'appliquerait pas ? C'est une question difficile à trancher pour nous, informaticien. Nous pouvons toutefois apporter quelques éléments en faveur du codage tabulaire pour les aires motrices. Premièrement, Ballard décrit des indices secondaires pour l'aire FEF du cortex frontal (cf. table 4.1), qui est une zone motrice. Deuxièmement, pour les aires somesthésiques, en miroir des aires motrices primaires, Miller affirme une équivalence d'organisation avec les aires visuelles [Miller *et al.*, 2001]. Enfin, et c'est dans cet argument que nous voyons la conciliation entre le possible caractère tabulaire du codage cortical moteur et les capacités d'extrapolation auxquelles Guigon se réfère, Jonathan Mink précise que le cortex produit une commande motrice hachée, qui consiste à déclencher un schéma moteur élémentaire [Mink, 1996]. Ce schéma, que Mink nomme *Motor Pattern Generator* (MPG), est réalisé par le cervelet et les ganglions de la base, pour obtenir une commande motrice lissée et ajustée d'un point de vue temporel. Un codage tabulaire des MPG est alors envisageable au niveau du cortex, les interpolations observées pouvant être le fruit de structures plus proches que le cortex de contrôleurs automatiques, comme le cervelet [Doya, 1999].

4.2.2 Multimodalité

La notion de modules constituant une architecture conduit à se poser la question de la gestion de la multimodalité au niveau du tissu cortical. Rappelons que pour nous cette question est d'autant plus intéressante que nous considérons que ces modules, en dépit du fait qu'ils soient concernés par des modalités différentes, sont de même nature, ce qui est un bel exemple de la généralité chère aux informaticiens. Nous avons également dit que la façon dont sont connectés les modules est pour nous une injection de connaissance, lorsque l'on construit des systèmes artificiels, et cette connectivité est, dans le cas de « vrais » cerveaux, une réponse de l'évolution au problème de gérer le comportement des animaux. Il paraît alors pertinent de considérer cette réponse de l'évolution pour injecter de la connaissance dans nos modèles artificiels, et c'est là que survient une mauvaise surprise, illustrée par la figure 4.6. La mauvaise surprise est que « *c'est le fouillis* », et que l'on serait bien en peine de construire une architecture artificielle basée sur ce genre de schémas de connexion. Il convient alors, pour notre entreprise, de s'intéresser à des descriptions plus fonctionnelles de la connectivité de l'architecture corticale, sachant que nos modèles, aujourd'hui, ne peuvent qu'esquisser des principes de connexion, et non reproduire fidèlement la connectivité du cortex. C'est là que la présentation du cortex par Yves Burnod prend pour nous tout son sens, et nous nous inscrivons de ce point de vue dans la démarche initiée par Frédéric Alexandre [Alexandre *et al.*, 1991].

Yves Burnod met particulièrement en avant la connectivité hypercubique de dimension 6 du cortex, comme nous l'avons vu page 77 (cf. la citation). Ce point de vue est celui d'une systématisme qui explique la complexité apparente de la connectivité corticale. Il en résulte qu'en chaque point du cortex se croisent six flux de connexions, chacun d'eux étant organisé en bandes [Burnod, 1989, page 161]. C'est ce qui a inspiré la connectivité via les *connection stripes* de *bijama*, que nous décrivons au paragraphe 4.5.2, qui est une connectivité systématique à partir de laquelle nous construisons des architectures complexes. La multimodalité semble être

donc assurée dans le cortex par le fait que sur sa surface « *se croisent* » des flux d'informations, qui se combinent et s'analysent au niveau des colonnes situées à l'endroit des croisements.

Lorsque l'on regarde plus globalement l'organisation de la modalité des informations sur la surface corticale, on constate que chacun des cortex gauche et droit est organisé en 5 pôles [Burnod, 1989, page 145] fonctionnels. Ces pôles, d'après Yves Burnod, sont :

- le pôle somato-moteur, situé au niveau du sillon central, qui traite des informations relatives au corps (toucher, position des articulations) et à la commande des muscles ;
- le pôle visuel ;
- le pôle auditif ;
- le pôle olfactif ;
- le pôle « *moléculaire interne* » , qui pilote la sécrétion d'hormones via l'hypothalamus.

Cette organisation en pôles décrit principalement le cortex postérieur, dont nous avons vu qu'il était en miroir avec le cortex frontal, qui « *reflète* » par conséquent lui aussi cette organisation modulaire. Le cortex frontal est en connexion avec des centres relatifs au plaisir et à la récompense, il est impliqué dans l'ordonnement de l'action [Fuster, 1997, chap. 8]. Nous avons d'ailleurs proposé un modèle informatique fonctionnel du cortex frontal durant nos travaux de thèse [Frezza-Buet, 1999; Frezza-Buet, 2002]. Notons enfin l'existence d'un pôle vestibulaire⁴⁷, relatif à la perception de la verticalité et de l'accélération du corps, particulièrement mis en avant par Alain Berthoz par exemple [Berthoz, 1997].

Ce qui est troublant, et particulièrement stimulant pour la recherche en informatique, est qu'entre ses pôles, au sein des régions intermédiaires entre ces différentes modalités, on observe des *gradients de modalités*. Ainsi le cortex pariétal, « *à mi-chemin* » entre le pôle visuel et le pôle somesthésique, gère la perception de l'espace de travail du corps, c'est le fameux axe « *where* » du traitement visuel. De même, le cortex temporal est à mi-chemin entre vision et le pôle moléculaire, et donne une couleur affective à ce qui est vu, ce qui, pour un être motivé et situé, revient à faire de la reconnaissance. C'est l'autre axe fameux du traitement visuel, l'axe « *what* » .

Les exemples de gradients de modalités sur la surface corticale que nous avons donnés sont certainement naïfs, mais il nous paraît que quelque chose de cet ordre se joue au niveau de la répartition pour le comportement dans son ensemble des ressources de calcul que sont les minicolonnes. Nous prendrons l'existence de cette notion de gradients comme hypothèse de travail, justifiant l'intérêt d'étudier l'allocation non supervisée de ressources indifférenciées pour la prise en charge de la multimodalité (cf. paragraphe 2.2.2 et figure 2.1).

4.2.3 La notion de *hiérarchie*

Qui dit architecture dit souvent hiérarchie de modules, surtout dans le domaine des sciences informatiques. Ce sur quoi nous nous attardons dans ce paragraphe est qu'en ce qui concerne l'*architecture corticale*, de nombreux points de vue de biologistes cautionnent également une vision en modules hiérarchisés. Nous séparons ces points de vue en deux grandes classes, la résonance et le codage prédictif, mais il est clair pour nous qu'il existe une intersection entre ces deux concepts.

47. Le système vestibulaire est un organe situé dans l'oreille interne, faisant office de centrale inertielle.

Résonance

La notion de *résonance* est un cas particulier de *relaxation* telle que nous l'avons présentée au paragraphe 3.6. Nous concevons par conséquent la résonance comme une *restitution de cohérence* spatiale, avec cette particularité qu'il s'agit d'une cohérence entre deux modules. La résonance a été introduite par Stephen Grossberg avec le modèle ART il y a vingt ans [Grossberg, 1987]. Ce concept est lié à la notion de hiérarchie car les deux modules ne jouent pas le même rôle.

Dans une relation de mise en résonance, un module neuronal « *bas niveau* » produit une distribution d'activités de ses unités, du fait d'une entrée externe par exemple. Ce profil d'activation est perçu par les unités d'un deuxième module, de « *haut niveau* », via des connexions *feedforward*. En conséquence de la stimulation *feedforward*, le module de haut niveau s'active. On parle de résonance lorsqu'il existe des connexions du module de haut niveau vers celui de bas niveau, visant à corriger l'activation du module de bas niveau en fonction de celle du module de haut niveau. Ces connexions *feedback* produisent un changement d'activité dans le module de bas niveau, et donc un nouveau profil d'activation est transmis via les liens *feedforward*. Nous décrivons ici un processus de relaxation qui stabilise, via le cycle des connexions *feedforward* et *feedback* les profils d'activités des deux modules.

Le rôle des modules n'est toutefois pas interchangeable. En effet, le module bas niveau est porteur d'une « *vérité* » qui alimente le module de haut niveau, dont la fonction est de reconnaître cette vérité. L'effet *feedback* consiste alors à amplifier certaines unités du module bas niveau, en inhiber d'autres, mais ne peut pas créer d'activité ex nihilo dans le module de bas niveau⁴⁸. Dans le modèle ART, la configuration stabilisée dans le module de bas niveau est comparée à l'entrée reçue, afin de voir à quel point la détection de haut niveau est capable de rendre compte de l'entrée. Cette comparaison pilote un processus de création de nouveaux prototypes dans le module de haut niveau, que nous ne détaillerons pas, mais dont on retiendra qu'il a été proposé par Grossberg pour résoudre le dilemme plasticité/stabilité dans les systèmes d'apprentissage [Grossberg, 1987].

Ce principe d'influence entre modules, même si l'on ne souhaite pas « *coller* » au équations du modèle ART, est général. Il a par exemple été utilisé en cascade pour modéliser les traitements visuels par la chaîne $LGN \rightarrow V1 \rightarrow V2$, pour montrer comment $V2$ peut restituer sur les LGN la connaissance qu'une image est constituée de lignes. Ce modèle, LAMINART [Grossberg, 1999], est à notre connaissance un des rares modèles à ne pas considérer les LGN comme une simple batterie de filtres, mais à bel et bien mettre en avant que cette structure expose les informations rétiniennes aux *feedback* du cortex.

Des auteurs comme Jean Bullier abondent dans ce sens, apportant au mécanisme de résonance l'argument de vitesses de transmission différentes dans le processus de relaxation [Bullier, 2001]. L'idée est alors qu'il existe une influence *feedforward* rapide qui prépare le module de haut niveau par une activation ciblée. Ce dernier produit rapidement un *feedback* plus diffus, destiné à préparer le premier module pour sélectionner ce qu'il va transmettre au second par des voies plus lentes. La notion de *feedforward* directif et *feedback* diffus qui préparent rapidement la chaîne de traitement, avant que l'information y transite plus lentement, débarrassée au passage du bruit par restitution de cohérence, est également évoquée dans [Friston, 2002], ainsi que dans un modèle appliqué à de vraies images [Schwabe et al., 2006], où l'on retrouve la constitution d'indices secondaires d'orientation de contrastes dans $V1$ (cf. table 4.1).

48. Le système n'hallucine pas.

Codage prédictif

La notion de *codage prédictif* [Rao and Ballard, 1999] est à notre sens le meilleur argument pour une organisation hiérarchique du cortex, dans la mesure où le rôle des modules de haut niveau et de bas niveau n'est pas interchangeable. Il s'agit selon ces auteurs de considérer que le rôle d'un module de haut niveau est de produire via les liens *feedback* une prédiction de l'activation du module de bas niveau. Cette prédiction inhibe le module de bas niveau, si bien qu'après soustraction de l'état du module de bas niveau avec ce qu'en prédit le module de haut niveau, n'est transmis par les liens *feedforward* que l'erreur qu'à commise le module de haut niveau. Cette erreur lui permet d'ajuster sa prédiction. La retransmission par les liens *feedback* de l'état du module de haut niveau est un *modèle génératif*, dont la fonction est de pouvoir générer ce que devrait être l'état du module de bas niveau. Le lien entre codage prédictif, modèles génératifs, et théorie de l'information est détaillé dans [Friston, 2002], ainsi que les conséquences de cette approche sur l'interprétation des activations du cortex observées à l'IRM fonctionnelle.

« The approach [predictive coding] postulates that neural networks learn the statistical regularities of the natural world, signaling deviations from such regularities to higher processing centers. This reduces redundancy by removing the predictable, and hence redundant, component of the input signal » .

Rajesh P. N. Rao and Dana H. Ballard [Rao and Ballard, 1999]

Le concept de codage prédictif peut être appliqué deux à deux aux modules de niveaux successifs dans une hiérarchie, comme l'illustre la figure 4.7-a. Le codage prédictif permet d'expliquer les phénomènes non classiques observés sur les champs récepteurs de neurones visuels, schématisés sur la figure 4.7-b. Il est à noter que des phénomènes analogues sont aussi étudiés avec LAMINART, ce qui soutient que codage prédictif et résonance sont des concepts qui se recouvrent.

Nous dirons pour conclure, avec Friston, qu'il y a un lien entre modèles génératifs, hiérarchie entre modules, résonance, et que ce lien en dit long sur la nature des traitements génériques réalisés par le cortex.

« [...] predictive coding in the context of hierarchical generative models not only accounts for many extra-classical phenomena seen empirically but also enforces a view of the brain as an inferential machine through its empirical Bayesian motivation » .

Karl Friston [Friston, 2002]

4.2.4 Discussion

Nous avons mentionné aux paragraphes 1.3.3 et 2.1.1 notre souhait de repenser la modularité des systèmes en s'appuyant sur ce qu'enseigne le cortex, dans la mesure où il nous apparaît que l'approche modulaire hiérarchique classique en informatique trouve ses limites quand il s'agit de spécifier une cognition située. Or de nombreux auteurs cautionnent une vision hiérarchique du cortex... Faut-il en conclure que le cortex fait la preuve qu'une architecture hiérarchique permet d'aborder la cognition située, et que la méthode classique d'ingénierie, en fait, convient parfaitement ?

Ce n'est pas si simple. Tout d'abord, remarquons qu'un auteur comme Jean Bullier, qui soutient la notion de hiérarchie dans l'organisation corticale, présente l'aire visuelle V1/V2, en bas de hiérarchie, comme un tableau noir sur lequel viennent se traiter toutes les composantes visuelles d'un comportement global, sous la sollicitation éventuelle d'autres modules.

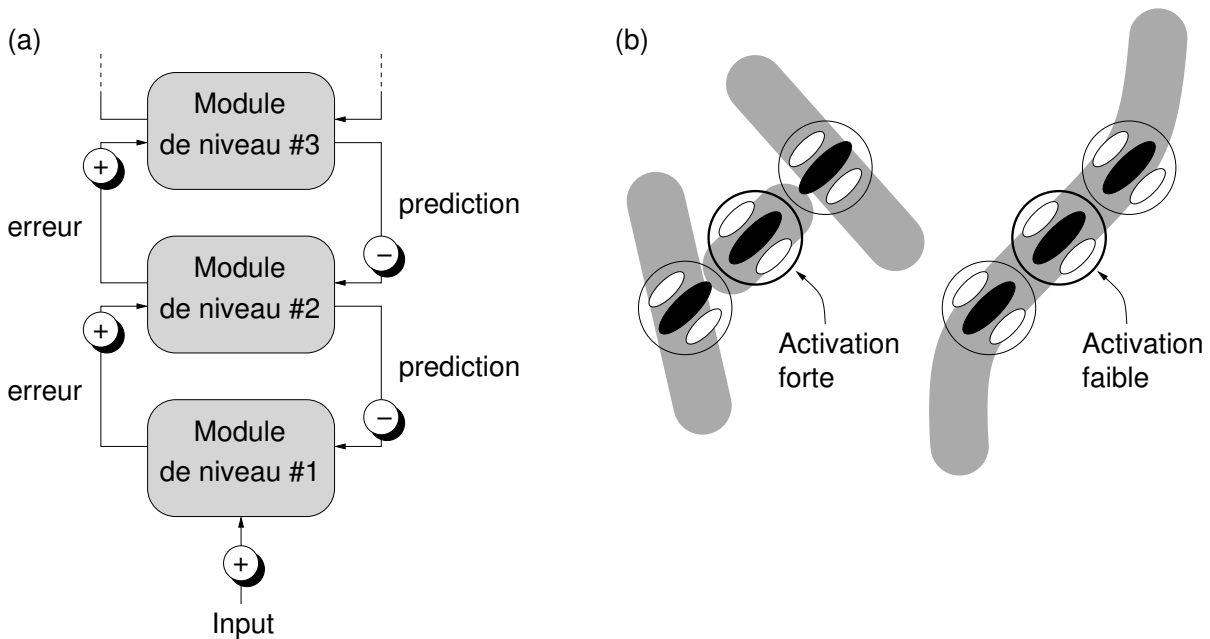


FIGURE 4.7 – Le codage prédictif dans une hiérarchie de modules (a), d’après [Rao and Ballard, 1999]. La figure (b) montre que la même unité peut répondre différemment, selon que son activité puisse être prédite (figure de droite) ou non (figure de gauche) par un module « supérieure » de détection de ligne.

« Through the rapid activation of feedback connections, the computations done at a more global level can be retroinjected into the general purpose areas V1 and V2 that act as active blackboards. » .

Jean Bullier [Bullier, 2001]

Autrement dit, Bullier nous présente V1/V2 comme le pôle intégré de la vision, ce qui nous fait apparaître cette aire non plus comme la première interface corticale, bas niveau, avec la stimulation visuelle, mais plutôt comme l’étage ultime d’extraction de la composante visuelle d’un comportement. Dans la continuité de ce raisonnement, nous préférons à la notion de hiérarchie, la notion de coexistence, sur la surface corticale, de différents pôles, comme le soulignent Burnod et Fuster [Burnod, 1989; Fuster, 1997]. Ces pôles, d’un point de vue fonctionnel, peuvent être interprétés comme autant de « regards » sur une même réalité, le comportement présent, et ces regards s’influencent, apportent chacun leur dose de restitution de cohérence, afin d’assurer un comportement situé adapté.

Les notions de codage prédictif et de dissymétrie des relations *feedforward* et *feedback* que nous avons mentionnées ne sont pas en contradiction avec ce point de vue, car la dissymétrie n’implique pas la hiérarchie. D’ailleurs les auteurs s’appuient souvent sur les relations « hiérarchiques » entre V1 et V2, et Burnod nous rappelle que la différenciation primaire/secondaire⁴⁹ est un des six axes de symétrie qui président à toute l’organisation corticale (voir la description des symétries page 77). Avançons ici qu’il est possible que la mise à disposition d’un modèle génératif pour régulariser l’activité d’une zone peut-être justement la fonction de cette dernière symétrie, et nous avons vu sur la figure 4.3 qu’il existe des relations spécifiques à cette fonction au niveau des couches de la minicolonne (sur cette figure, (e) est le module prédictif de haut

49. V1 et V2 sont symétriques par rapport à l’axe primaire/secondaire.

niveau, et (f) le module recevant le signal), ces relations n'étant qu'un exemple de celles, plus générales, qui semblent exister au niveau intra-cortical.

Depuis nos travaux de thèse [Frezza-Buet, 1999], nous avons rejeté la notion de hiérarchie, et ce rejet a été constructif. En effet, dans ces travaux, nous avons montré comment des contraintes (la notion d'appel de Burnod) pouvaient circuler via les cartes primaires et associatives pour provoquer le comportement, sans avoir recours à des modules superviseurs, au sein desquels aucune généralisation ne peut se produire. C'est ce rejet de la hiérarchie des modules qui nous a permis d'aborder l'extraction par chaque carte de la compétence comportementale qui lui revient, ce qui est une forme de généralisation partielle [Frezza-Buet and Alexandre, 2002]. Depuis ces travaux, même si notre approche de « *l'algorithmique corticale* » a beaucoup changé, sous l'influence en particulier des champs neuronaux dynamiques, nous n'avons pas remis en cause ce rejet de la notion de hiérarchie, ce qui ne nous empêche pas de considérer avec le plus vif intérêt les travaux de Friston, Ballard, Bullier et d'autres, qui ne sont finalement pas contradictoires avec notre prise de position. Simplement, en biologie, invoquer une hiérarchisation procède d'une démarche, ô combien louable pour nous, de donner un sens à l'organisation corticale, et il est clair que les arguments contre la hiérarchie que nous invoquons sont du ressort de l'informatique. Ces arguments font partie de ce que nous proposons comme réflexions, en tant qu'informaticien, à la compréhension du rôle des calculs corticaux dans le traitement situé de l'information.

4.3 Dynamique fonctionnelle du cortex dans son ensemble

Nous avons tenté, dans notre présentation des modules corticaux tout comme dans celle de l'architecture dans laquelle ils interviennent, de soutenir l'idée que le cortex réalise *une* fonction qui est, à des variantes d'un second ordre près, la même en chaque point de sa surface. Nous allons dans ce paragraphe donner quelques éléments sur l'établissement de la spécialisation des aires corticales, que nous voyons comme un processus d'ajustement dynamique de la fonction générale de la surface corticale à l'information qui y est traitée. Notons toutefois que dès les années 80, c'est bien comme un processus dynamique de construction que Dana Ballard décrit le rôle du cortex.

« *A major function of the cortex is to compute collections of invariants at different levels of abstraction* » .

Dana H. Ballard [Ballard, 1986]

Nous aborderons également la question de la dynamique globale du cortex, non plus sous l'angle de la structuration des fonctions, mais sous celui de la coordination des activations.

Nous ne donnerons de ces deux points que quelques grandes lignes, tant les ressorts de la dynamique globale du cortex sont encore aujourd'hui mal connus, et peu abordés par les modèles.

4.3.1 Une organisation globale non supervisée

L'organisation globale de la surface corticale répond à de nombreux paramètres, mais il est intéressant de constater que les auteurs qui la décrivent ne manquent pas de faire référence à l'extrême plasticité de cette organisation dans les cartes dites « *homotypiques* » [Mountcastle, 1997; Jones, 2000]. Certains auteurs proposent de considérer que les mécanismes de différenciation sont, à l'instar de la dominance oculaire dans V1, le fruit de l'exploitation d'instabilités de Turing dans la dynamique du substrat cortical [Oster and Bressloff, 2006], comme l'illustre la figure 4.8.

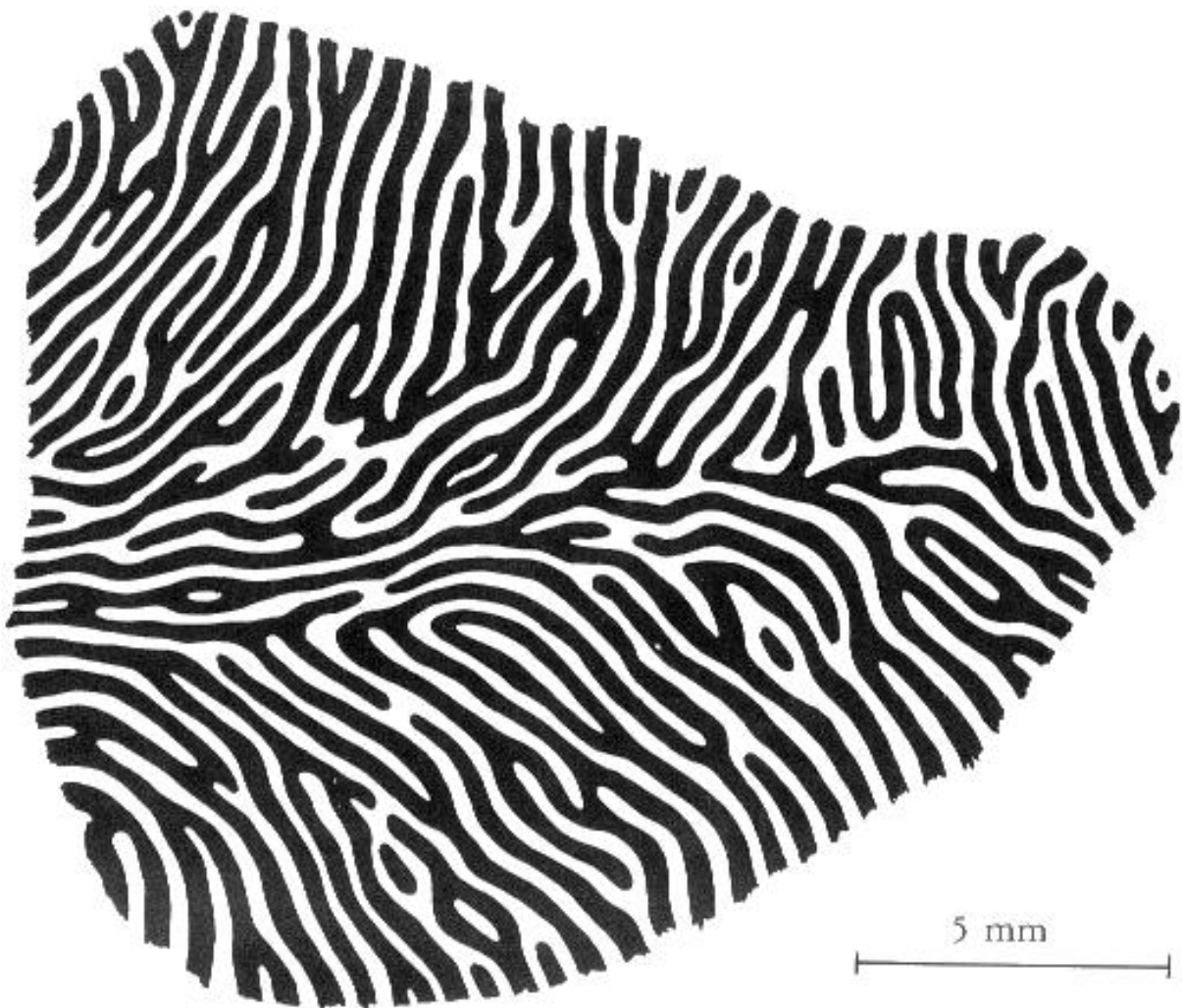


FIGURE 4.8 – Bandes de dominance oculaire (en noir pour un œil, en blanc pour l'autre) sur la surface de l'aire visuelle V1 (d'après [Hubel, 1994]). Ce motif suggère la présence d'une instabilité de Turing, comme celles illustrées sur la figure 3.8.

Cette plasticité s'exprime dès la genèse du cortex, qui semble bien plus dirigée par une adaptation à des signaux divers, comme des gradients de concentration de substances chimiques, qu'à un câblage « *en dur* » qui serait piloté par la génétique [Sur and Rubenstein, 2005]. Durant la maturation du cortex, la répartition des surfaces accordées aux différentes aires peut varier. Nous renvoyons le lecteur à [Elbert and Rockstroh, 2004] pour une revue de ce problème, dont nous ne présentons ici que quelques morceaux choisis. Chez les personnes lisant le Braille, la répartition sur les aires somesthésique des doigts de la main diffère des sujets qui ne pratiquent pas cette lecture, l'implication de l'exploration du texte Braille étant une activité très particulière des doigts. Il en est de même des aires relatives à la main gauche chez les guitaristes⁵⁰ droitiers. De même, chez les personnes aveugles, on a constaté que l'aire auditive était 1.8 fois plus grosse que chez les voyants, et que cette aire est également plus développée chez les musiciens. Il semble donc que cette plasticité s'observe aussi bien chez des personnes ayant une différence dès la naissance (cécité par exemple), que chez des personnes qui, du fait d'une pratique particulière, spécialisent leur comportement (musiciens). Il semblerait qu'il y ait deux niveaux de plasticité, le premier qui est requis pour la mise en place du système nerveux par l'expression de quelques gènes seulement, et le second qui est requis par la spécialisation de ce système pour répondre à l'environnement du sujet. Ce dernier reste plastique chez les sujets adultes.

Dans [Elbert and Rockstroh, 2004] toujours, les auteurs donnent quelques principes de réorganisation corticale :

- En cas de déafférentiation (perte d'un membre par exemple, lésion d'une partie de la rétine), les aires voisines à la zone qui n'est plus « *alimentée* » l'envahissent progressivement, ce que nous interprétons comme un bel exemple de réutilisation d'une surface de calcul devenue inutile.
- Il y a un rôle central du sommeil paradoxal dans la plasticité globale de la surface corticale.
- La plasticité ne survient que lorsqu'il y a une composante motivationnelle forte du comportement qui la requiert. Une étude a été faite sur des singes, en les plaçant dans une situation où ils devaient, pour les uns, faire une discrimination tactile fine pour obtenir une récompense, alors que pour les autres, c'est une discrimination auditive qui est requise. Les aires correspondantes se sont différenciées significativement selon les groupes, ce que nous interprétons là aussi comme un bel exemple du caractère situé des mécanismes de plasticité cérébrale⁵¹.

Notons enfin que cette réorganisation peut-être vécue douloureusement par les sujets victimes d'amputation, qui souffrent au niveau de leur « *membre fantôme* ». La pose d'une prothèse, dont la manipulation invoque l'aire initialement déafféree, réduit la douleur ressentie. Ces mécanismes de réorganisation sont au cœur d'études médicales sur la rééducation de patients, études que nous ne ferons que mentionner ici.

Notons enfin que l'on trouve dans la littérature des analyses temporelles de la dynamique de cette plasticité corticale, dont certaines font état d'échelles extrêmement courtes, de l'ordre de quelques heures [Stavrinou *et al.*, 2007]. Ces études se basent sur un paradigme classique, qui consiste à attacher entre eux les doigts d'un humain, et observer l'effet de cette limitation de mouvement sur l'organisation des aires corticales relatives à la main, par des techniques de magnétoencéphalographie.

50. La pratique de la guitare suppose pour un guitariste droitier d'avoir une main gauche très agile.

51. Cette remarque pourra sembler évidente au biologiste, mais elle soulève une question difficile pour qui cherche à concevoir un mécanisme de plasticité en informatique.

4.3.2 Un parallélisme fonctionnel cortical ?

À un autre niveau que celui de la flexibilité des processus d'auto-organisation, la dynamique fonctionnelle du cortex pose, et c'est étonnant de la part d'une structure neuronale, la question du parallélisme des traitements effectués. Que le cortex, comme le cerveau en général, soit une structure de calcul parallèle ne fait aucun doute, même si l'on peut discuter de la granularité de ce parallélisme (la synapse, le neurone, la microcolonne, etc.). Ce qui est plus délicat est de savoir dans quelle mesure le cortex est capable de faire plusieurs choses en même temps, et c'est la question du parallélisme fonctionnel que nous posons dans ce paragraphe.

Supposons que l'étalement sur la surface corticale des différentes modalités soit le reflet d'une gestion, par le cortex, de plusieurs objets en parallèle. Se pose alors le problème suivant, décrit entre autres par Ballard [Ballard, 1986], dit *problème du binding*.

Lorsqu'un sujet perçoit une image, sur laquelle se trouvent par exemple un triangle rouge à droite et un cercle bleu à gauche, la détection des positions des deux objets s'effectue dans les aires pariétales, alors que les couleurs sont traitées en V4. Dit autrement et *très schématiquement*, en V4, les colonnes détectant le rouge et les colonnes détectant la présence de bleu sont activées conjointement, et dans les aires pariétales, on a l'activation de deux colonnes, une pour la position de chaque objet. Le problème du *binding* est alors de savoir comment le cortex réussit à ne pas associer de façon erronée la position et la couleur, en considérant par exemple qu'il y a un objet rouge à gauche. Comment les activités de V4 sont reliées à celles des aires pariétales pour que, malgré l'éclatement des modalités, les propriétés relatives à un objet restent fonctionnellement liées ? Ce problème laisse apparaître un inconvénient à la gestion éclatée de l'information par le cortex, toutefois compensé par un autre mécanisme qui reste à identifier. Edelman et collègues en proposent un, qui est la synchronisation des phases des potentiels d'action des unités relatives à un même objet [Seth *et al.*, 2004].

En général, le *binding* est abordé dans les modèles à propos de la modalité visuelle, et les cartes corticales de ces modèles ont des indices primaires correspondant à l'image perçue. Or la vision réelle, en tant que perception, est non seulement « *déformée* » du fait de la magnification de la vision centrale au détriment de la vision périphérique, mais elle est surtout, comme toute perception, *motrice*, et par conséquent articulée dans le temps, comme nous l'avons défendu au paragraphe 1.3.2. Si l'on en revient à cette conception de la perception, on se rend compte que le problème du *binding* disparaît, comme le fait d'ailleurs remarquer Kevin O'Regan [O'Regan and Noë, 2001, section 8.3]. Si la vision est effectivement palpation par le regard, à un instant donné, le cortex traite, pour ce qui est de la vision, le contour, la couleur, la position par rapport au corps, la texture, etc... de l'*unique* objet visé, et nul besoin de mécanisme de *binding* puisque nécessairement, tout ce qui est éclaté sur la cortex ne concerne que l'objet visé à cet instant. Le fait de gérer plusieurs objets, vient de notre capacité à rester cohérent lorsqu'on les *saisit tour à tour*, et cette gestion sérielle renvoie à la question que nous avons déjà évoquée de notre capacité à articuler dans le temps des raisonnements imbriqués, capacité qui est selon Fuster le propre du *pôle* moteur du cortex, le cortex frontal (cf. page 20).

Rejeter la question du *binding* revient, pour nous, à être cohérent avec ce que nous avons soutenu de la perception au chapitre 1, et évite de devoir introduire des mécanismes supplémentaires dans l'interprétation des fonctions du cortex, en accord avec le principe de rasoir d'Occam.

Nous partons donc de l'hypothèse que le cortex, quoique parallèle, est une structure dédiée à sérialiser les informations qui concernent le sujet, de sorte à produire un comportement qui s'articule dans le temps. Le corollaire en est qu'à un instant donné, cette structure pourtant massivement parallèle réalise la prouesse de ne traiter qu'un seul problème, dont elle appréhende sur sa surface un maximum de facettes. Et si le cœur de toute la fonction corticale était tout simplement de ne traiter qu'un problème à la fois ?

4.4 Le calcul cortical

Nous proposons dans ce chapitre de cerner ce que pourrait être en informatique un calcul dit « cortical ». Il ne s'agit toutefois pas d'en donner une définition, le domaine n'étant à notre sens pas assez mûr pour cela, mais plutôt d'exhiber quelques propriétés, qui sont autant d'arguments, lorsqu'elle sont vérifiées par un système, pour le qualifier de « cortical ». Nous ferons, une fois ces propriétés énoncées, la proposition d'un système cortical, qui est celui auquel nos recherches ont mené jusqu'à aujourd'hui, et sur lequel s'appuieront les développements futurs de nos travaux.

4.4.1 Propriétés « corticales »

La description que nous avons faite du cortex est biaisée par le regard d'informaticien que nous portons sur son organisation, elle n'a sans doute pas l'objectivité que pourrait attendre un lecteur biologiste, tant il est vrai que nous n'avons, par exemple, pas présenté d'arguments en défaveur de la généricité des minicolonnes lorsque nous avons soutenu ce point. Ce regard partial sur le cortex est justement ce que nous pensons apporter aux sciences cognitives, et c'est bien entendu lui qui guide la conception de nos modèles informatiques. Les liens que nous voyons entre le cortex et les propriétés informatiques que nous allons énoncer sont argumentés dans ce chapitre par l'ensemble de la description orientée que nous avons proposée jusqu'ici, et le lecteur qui n'adhérerait pas à ce rapprochement pourra toujours se contenter de voir dans ces propriétés une description des architectures informatiques que nous étudions, gardant en mémoire que leur motivation n'est pas de modéliser le cortex, mais est celle que nous avons énoncée au paragraphe 2.2.

Local mais non localiste

Le calcul réalisé au sein d'un système cortical doit, selon nous, être *strictement local*. Dit dans un langage plus informatique, cela signifie que les instances des unités de calcul, les grains fins, n'ont accès à aucune variable globale pour réaliser leur calcul, à l'instar des automates cellulaires par exemple (cf. paragraphe 3.3).

Cette exigence est celle qui a guidé nos travaux de thèse qui, s'appuyant sur le modèle Carl proposé par Pascal Blanchet [Blanchet, 1992; Blanchet, 1994; Blanchet and Alexandre, 1995], les ont prolongé en particulier dans cette direction [Frezza-Buet, 1999; Frezza-Buet and Alexandre, 2002]. Toutefois, et c'est certainement le principal point que nous avons remis en cause depuis ces travaux, un calcul cortical nous apparaît devoir s'affranchir d'un localisme trop prononcé, ce que nous allons préciser.

Le *codage tabulaire* que nous avons défini page 4.2.1 suggère qu'une minicolonne représente un quantum d'information, mais nous pensons que ce quantum doit rester petit, ce qui est le cas dans les exemples d'indices secondaires que nous avons cités. On pourrait toutefois imaginer que,

dans des aires plus « *abstraites* », les quanta soient des représentations plus symbolique, et l'on arrive ainsi à la notion de *neurone grand-mère*⁵², à savoir la minicolonne qui représente dans son cortex un percept aussi complexe que la grand-mère du sujet. Une approche *localiste* est dans ce contexte une approche qui soutient que des neurones de cette sorte, à savoir relatifs à une *représentation* très intégrée, existent effectivement. Or nous avons rejeté le représentationnalisme dans notre appréhension de la cognition située (cf. paragraphe 1.3.1). Le modèle Carl, ainsi que celui que nous avons proposé durant notre thèse, sans être aussi extrêmes que le neurone grand-mère, ont ce caractère localiste, et se ramènent à une gestion connexionniste et distribuée de représentations presque symboliques.

La subtilité d'un modèle cortical, selon nous, tient à pouvoir s'appuyer sur un calcul strictement local, sans céder au localisme. Ainsi, du fait qu'il n'y a pas de localisme, les unités locales doivent pouvoir collectivement représenter un concept, ce qui amène à se confronter au problème de l'émergence d'une cohérence répartie et non supervisée. Ce problème est à notre sens bien plus générateur de paradigmes nouveaux pour l'informatique qu'une manipulation explicite de symboles.

Calcul surfacique à grains fins homogènes

Un système ayant des propriétés de calcul corticales se doit, selon nous, de reposer sur une répartition des calculs dans un espace bidimensionnel. Cette répartition se réalise par la répartition, sur cet espace, d'une collection d'unités de calcul *homogènes*. Le système ainsi formé est un cas particulier de système parallèle à *grain fin*.

Nous affirmons donc le caractère surfacique du calcul comme une propriété majeure pour qu'un système prétende à être d'inspiration corticale. Ceci s'entend, toutefois, si cette organisation en surface est effectivement exploitée par le système. Nous avons parlé dans les paragraphes précédents (cf. paragraphe 4.3) d'un processus de relaxation pour réduire l'information « *active* » sur la surface. Pour nous, il apparaît que la surface du cortex est justement la dimension suivant laquelle se réduit localement l'information, par un processus de *compétition*, ce qui est le mécanisme élémentaire de la fonction de restitution de cohérence que nous avons attribué au cortex.

Calcul multimodal

La propriété que nous avançons pour cerner ce qu'est le calcul cortical est l'aptitude à gérer et combiner des informations de modalités diverses. Ainsi, en plus d'avoir à gérer la compétition que nous avons décrite au paragraphe précédent, la minicolonne doit gérer le fait qu'elle est, par ses connexions vers d'autres modules, au carrefour de plusieurs flux d'information. Selon nous, c'est la combinaison de cette gestion multimodale avec les processus de compétition localisés sur la surface qui produit une décision multimodale cohérente, prise collectivement. Les « *relaxations surfaciques* » que nous avons mentionnées se doivent donc de participer à une relaxation plus globale du système, lui assurant une gestion multimodale et organisée du comportement.

Apprentissage en ligne non supervisé

Enfin, s'il est une notion séduisante dans le calcul cortical, c'est certainement celle de l'adaptativité de ce substrat, ce qui amène la question de l'apprentissage. Cette question est transverse aux propriétés précédentes, puisqu'à part celles qui font la promotion d'un calcul à grain fin non

52. Il semblerait que l'on doive ce terme à Jérôme Lettvin.

supervisé, les autres sont des propriétés que le système doit acquérir pour adapter sa gestion de la cohérence, sa gestion de la multimodalité, à l’environnement dans lequel il se trouve.

Nous dirons que cet apprentissage, pour être cohérent avec notre vision du calcul cortical, se doit d’être non supervisé, et qu’il doit surtout s’effectuer en ligne. Il est en effet peu concevable, pour un système situé, de devoir être tributaire d’un séquençement en phase d’apprentissage puis en phase d’exploitation. Or la mise en place d’un mécanisme d’apprentissage permanent, et *on-line*, soulève des difficultés en informatique, ce qui fait aussi son intérêt.

4.4.2 Méthodologie de construction des modèles

Face à la complexité informatique de la conception d’un système réalisant un calcul cortical, il convient d’adopter une méthodologie. Certains auteurs, comme notamment Philippe Gaussier et ses collègues, choisissent de montrer qu’on peut rendre des architectures neuronales opérationnelles sur des robots, leur confèrent même des comportements que l’on peut qualifier de cognitifs. La méthodologie est ici de s’appuyer sur la faisabilité robotique pour garder la garantie de systèmes situés. On citera par exemple [Gaussier *et al.*, 2002], qui réalise une tâche de navigation par un modèle dont certaines propriétés se rapprochent de celles que nous avons énoncées au paragraphe précédent, mais aussi [Toussaint, 2006] qui propose une méthode proche de celle de Gaussier, avec une affirmation plus nette d’une modélisation corticale.

Une autre méthodologie, qui est celle que nous suivons, consiste à partir de la définition d’un système de calcul, et d’en étudier les propriétés. C’est également ce que proposent Nicolas Rougier et ses collègues [Rougier and Vitay, 2006; Rougier, 2006; Fix *et al.*, 2007], pour qui le calcul cortical est assuré par des modules qui sont des champs neuronaux dynamiques. Cette méthodologie, qui consiste à aborder le problème par la définition d’un modèle d’exécution, est également partagée par les auteurs qui proposent des simulateurs, ce qui est aussi notre cas. En effet, proposer un simulateur suppose d’offrir une large palette d’utilisations, tout en confinant ces utilisations dans un cadre, qui permet justement de prendre en charge automatiquement les problèmes de simulation (parallélisation, visualisation, etc...). On trouve ainsi la définition de modèles d’exécution pour des simulateurs corticaux dans [Rice *et al.*, 2007], qui repose sur une approche bayésienne pour le codage prédictif, mais aussi dans [Ananthanarayanan and Modha, 2007; Djurfeldt *et al.*, 2008] pour le simulateur BlueBrain, visant à tourner sur les supercalculateurs BlueGene d’IBM.

Le paragraphe suivant présente le modèle d’exécution `grumpf` que nous avons défini, et pour lequel existe un simulateur (cf. paragraphe 9.6). La spécification de ce modèle répond à la nécessité d’abstraire des modèles pour en proposer un simulateur, alors que la définition du modèle d’exécution `bijama` que nous présentons également dans ce chapitre répond, elle, à la méthodologie de conception au préalable d’un modèle pour en étudier les propriétés. `grumpf` et `bijama`, qui sont la colonne vertébrale des recherches que nous avons menées, sont deux arguments pour asseoir la pertinence, dans notre contexte, d’une méthodologie centrée sur la définition de modèles informatiques.

4.4.3 Le modèle `grumpf`

Le modèle informatique `grumpf`⁵³ spécifie un type de calcul particulier, que nous allons détailler dans ce paragraphe. En pratique, il s’agit d’une bibliothèque C++ qui fournit les classes et fonctions nécessaires pour développer une application respectant ce type de calcul, accompagnée de nombreux utilitaires. Nous renvoyons le lecteur désireux de plus de détails pratiques au

53. `graphical utilities for the modelling of parallel functions`.

paragraphe 9.6, nous ne décrivons ici que le type de calcul lui-même. Le modèle `grumpf` est impliqué dans des collaborations avec Stéphane Vialle de notre équipe, ainsi qu’avec Jens Gustedt de l’équipe Algorille du Loria, puisqu’il est instancié sous la forme d’une architecture logicielle parallèle (cf. paragraphe 8.1.2).

Un modèle connexionniste

Le modèle `grumpf` est une extension des automates cellulaires et des CNN (cf. paragraphes 3.3 et 3.3.3). Comme eux, ce modèle est basé sur des unités de calcul élémentaires. Une *unité* est constituée d’un nombre arbitraire, mais constant, de *potentiels*, qui sont des variables réelles. Le calcul réalisé par une unité consiste à *mettre à jour* l’ensemble de ses potentiels.

Les unités d’un calcul répondant au modèle `grumpf` sont organisées en un réseau. Il s’agit d’un graphe orienté, dont les nœuds sont les unités, et les arêtes des *liens*. Une arête $A \leftarrow B$ de l’unité B vers l’unité A est hébergée au niveau de l’unité A , B n’en a aucune visibilité. Cette arête correspond à un droit de lecture pour A des potentiels de l’unité B .

Nous avons souhaité, lors de la conception du modèle, mettre en avant une notion de *connexions typées*. Le modèle impose pour cela de regrouper les liens qu’héberge une unité dans des ensembles, tous les liens d’un même ensemble étant par définition des liens de même type. Ce regroupement permet d’implémenter des règles de mises à jour utilisant par exemple « *La somme des valeurs reçues par les liens de type 1, multipliée par la somme des valeurs reçues par les liens de type 3, etc.* ». La figure 4.9 résume le modèle connexionniste que nous venons de décrire.

Les mises à jour

Alors que les potentiels d’une unité sont des variables réelles⁵⁴, la mise à jour de cette unité est un *événement* discret dans le temps. Toutes les unités du modèle n’ont pas nécessairement la même règle de mise à jour, elles ne possèdent d’ailleurs pas toutes le même nombre de types de lien (i.e. d’ensembles de liens), ni même le même nombre de potentiels. Toutefois, quelle que soit l’unité, une mise à jour consiste à modifier ses potentiels en fonction de toutes les valeurs qui peuvent être lues (ses propres potentiels et ceux consultables via les liens), et d’éventuellement un état interne.

Dans le modèle `grumpf`, nous avons pris le parti d’imposer qu’une unité ne peut subir deux mises à jour consécutives sans que toutes les autres aient effectué une mise à jour. On appelle un *pas d’exécution* l’application d’une seule mise à jour à toutes les unités. L’évolution du système consiste à enchaîner des pas d’exécutions successifs.

Les pas d’exécutions du modèle peuvent, selon ce que spécifie le concepteur, s’effectuer soit de manière synchrone, soit de manière asynchrone, ces notions étant celles définies au paragraphe 3.3.2. En pratique, surtout dans le cas d’une exécution sur un cluster de machines, l’implémentation peut se réserver le droit de ne pas assurer un asynchronisme parfait, mais ces aspects ne sont pas à prendre en charge au niveau de l’utilisation de la bibliothèque `grumpf` qui, grâce aux collaborations avec les chercheurs en parallélisme cités plus haut, masque ces difficultés.

54. Du moins autant que puissent l’être les nombres à virgule flottante sur un microprocesseur.

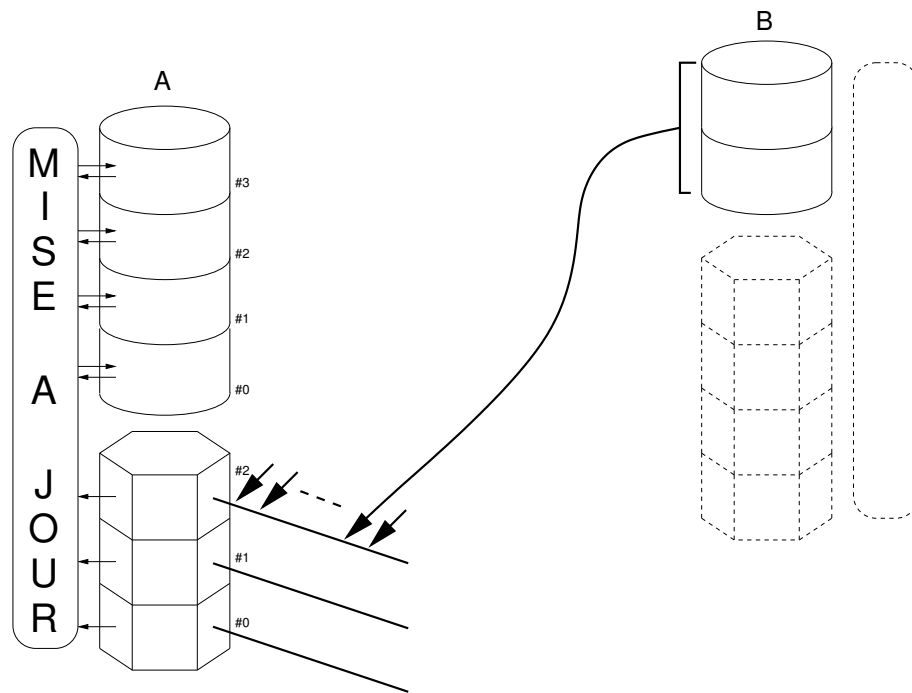


FIGURE 4.9 – Le modèle informatique **grumpf** est un modèle connexionniste. Les potentiels sont représentés par des portions cylindriques, et les ensembles de liens par des portions hexagonales. Les liens hébergés par une unité, qui sont des autorisations de lecture des activités d’une autre unité, sont regroupés dans des ensembles afin d’induire un typage des connexions du modèle. *A* est l’instance d’une classe d’unités ayant 4 potentiels et 3 types de liens, alors que *B* est l’instance d’une classe d’unités ayant 2 potentiels et 4 types de liens. Parmi les liens de type #2 de *A* (liste de lien étiquetée #2 sur la figure), il y en a un qui permet de lire *tous* les potentiels de *B*.

Construction d'une architecture connexionniste

La conception d'une architecture connexionniste à l'aide du modèle `grumpf` se décompose en une succession d'étapes.

La première étape consiste à définir des classes⁵⁵ d'unités. Chaque classe spécifie un nombre de potentiels, de types de liens, éventuellement des variables d'état interne, et enfin une fonction de mise à jour.

La deuxième étape consiste à définir des classes de liens, dans la mesure où l'on peut étendre⁵⁶ la simple fonction de lecture assumée par les liens, en ajoutant par exemple un poids.

La construction effective du modèle commence effectivement à la troisième étape, où l'on alloue des instances des classes d'unités précédemment définies. Afin de référencer ces unités par la suite, et uniquement à cette fin, elles sont rangées dans des *cartes*. Une carte, dans ce modèle, est une collection d'instances d'unités, dont les éléments sont indexés par un couple d'entiers⁵⁷. Une unité est repérée dans le système par le nom de la carte dans laquelle elle se trouve, et par ses coordonnées entières 2D dans cette carte. Dans ce modèle, toutes les positions disponibles dans une carte pour stocker des unités ne sont pas forcément utilisées, rien ne garantit que toutes les unités d'une même carte soient des instances de la même classe, et l'appartenance d'unités à telle ou telle carte ne préjuge en rien de la connectivité.

Une fois les cartes créées et les unités allouées puis « rangées » dans ces cartes, ce sont les liens qui sont créés. Ils peuvent être des instances de classes de lien différentes, même s'ils sont insérés dans un même ensemble/type d'une unité. À la fin de cette étape, le graphe constitué par les unités et leurs liens est créé, et, encore une fois, la répartition des unités sur différentes cartes ne contraint pas ce graphe, les deux opérations sont indépendantes. Dit autrement, *la notion de carte n'intervient pas dans les calculs*.

La dernière étape consiste à « lancer les calculs », c'est-à-dire à déclencher une succession de pas d'exécutions.

L'implémentation que nous avons réalisée de ce modèle permet à qui veut réaliser un calcul de ce type d'obtenir un logiciel qui est un serveur, au sens TCP/IP du terme. Ce serveur offre à des clients, que nous fournissons, la possibilité de contrôler l'exécution du calcul, et de visualiser en ligne l'état du système. Cette visualisation se faisant en projetant sur la surface d'un écran les potentiels, dont les numéros sont choisis par le client, des unités d'une carte. C'est l'unique intérêt des cartes dans ce modèle. La figure 4.10 donne un exemple du type de clients graphiques aujourd'hui disponibles.

Généricité du modèle

Le modèle `grumpf` est une extension encore très généraliste des automates cellulaires et des CNN. Nous avons défini ce modèle pour l'implémentation de nos modèles, et particulier du modèle `bijama` décrit au paragraphe suivant. Dans ce cas effectivement, nous ferons correspondre la notion de cartes du modèle `grumpf` à la notion de module cortical 2D, implémenté dans `bijama`. Toutefois, la généricité de `grumpf` a permis de l'utiliser dans un tout autre contexte, celui de la résolution numérique d'équations aux dérivées partielles (cf. paragraphe 6.1). C'est autour de la généricité de ce modèle, du fait qu'il soit assis sur une collaboration solide avec chercheurs en parallélisme, qu'ont pu se réunir physique des plasmas, nanotechnologies, physique des

55. Il s'agit effectivement de classes, au sens C++ du terme, en pratique.

56. Avec notre bibliothèque C++, cette extension est réalisée par héritage.

57. C'est en pratique un tableau à deux entrées.

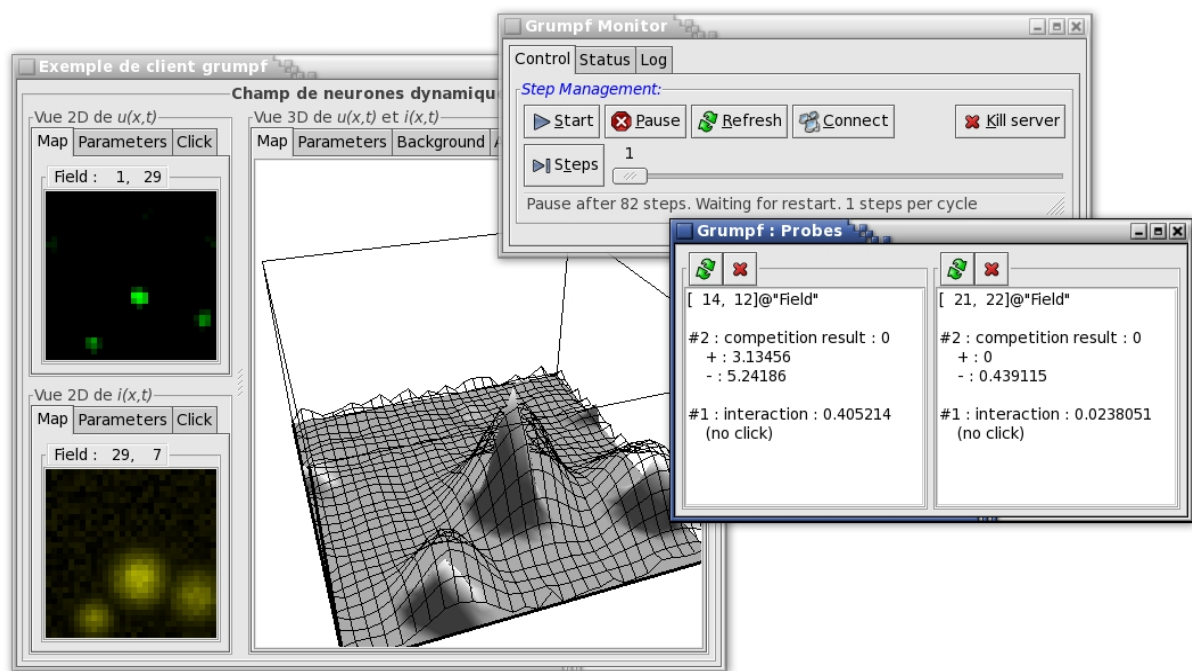


FIGURE 4.10 – Visualisation par un client graphique d’un serveur **grumpf** qui implémente l’équation 3.6 page 55. La fenêtre de gauche est un client graphique contenant des visualisateurs 2D et 3D, celles du haut est un client graphique permettant l’exécution pas à pas, la sauvegarde et la restauration de l’état du système, etc. La fenêtre de droite permet à l’utilisateur d’obtenir un message précis pour quelques unités, à des fins de mise au point par exemple.

matériaux photo-réfractifs, neurosciences computationnelles, dans le cadre du projet `InterCell` (cf. paragraphe 8.1.2).

Dans ces contextes, l'apport de `grumpf` réside dans l'interactivité et la visualisation. Pour les physiciens par exemple, il est important de visualiser la convergence en cours afin de détecter avant la fin d'une longue simulation des erreurs de modélisation. Pour nous qui programmons des systèmes distribués à grain fin, le retour visuel est essentiel pour maîtriser la complexité d'effets émergents, difficilement anticipables par une analyse mathématique préalable (cf. paragraphe 2.1.1). De plus, l'interactivité de `grumpf`, que nous ne détaillerons pas ici mais qui est argumentée dans le projet `InterCell`, permet de créer une interaction permanente entre le calcul en cours et un système externe, comme un robot, ce qui autorise l'utilisation de `grumpf` pour la mise en œuvre de systèmes situés. Initialement conçu pour machines à mémoire partagée, de type multi-processeurs, le modèle `grumpf` est aujourd'hui porté sur cluster de PC, ce qui est un des résultats du projet `InterCell`. Ce portage est présenté au paragraphe 9.7.

4.5 Le modèle *bijama*

Le modèle `bijama` (Biologically Inspired Joint Associative Maps) a été mis au point au cours de la thèse d'Olivier Ménard [Ménard, 2006], et constitue un « *incrément* » significatif de nos travaux de recherche, incrément qui nous permet d'entrevoir des solutions aux objectifs que nous avons formulés au paragraphe 2.2.

Ce modèle est une spécification du modèle informatique `grumpf` que nous avons présenté, spécification très orientée vers la définition d'un *calcul cortical* au sens où nous l'avons défini au paragraphe 4.4. La propriété d'être un système *local* et à *grains fins* est assurée par `grumpf` lui-même, et `bijama` ajoute le caractère *surfactive* des calculs, la *multimodalité*, ainsi qu'un apprentissage effectivement *en ligne* et *non-supervisé*. Ce modèle a été publié et nous renvoyons le lecteur à [Ménard and Frezza-Buet, 2005] pour le détail des formules, ainsi qu'à la thèse d'Olivier Ménard [Ménard, 2006].

4.5.1 Organisation des calculs en « *étages* »

Le modèle `bijama` est un modèle informatique respectant des principe de calcul cortical. Le caractère informatique du modèle s'exprime par la volonté de systématiser les connexions et les calculs, afin de pouvoir exprimer simplement des architectures impliquant de nombreux modules. Cette simplicité s'accompagne de l'automatisation de la conception du modèle à différents niveaux.

C'est dans cet esprit que les calculs réalisés au niveau des unités de `bijama` sont organisés en *étages*. Rappelons que les unités `grumpf` gèrent la mise à jour de variables, les *potentiels*, et l'organisation en étages structure cette mise à jour.

Premièrement, le modèle impose que toute unité n'expose aux unités qui ont un lien vers elle qu'un seul potentiel, ce qui est une restriction par rapport à `grumpf`. Ce potentiel, qui dans notre implémentation est le potentiel numéro 0 (référéncé #0 sur les figures à venir), exprime de degré d'activation de l'unité. Cette activité globale, que nous notons A^* , est le résultat du calcul de l'unité, calcul auquel participent tous les étages.

Prenons l'exemple d'un champ neuronal dynamique, tel que nous l'avons décrit au paragraphe 3.6.2, utilisant l'équation 3.6, pour poursuivre la présentation de la notion d'étages. L'unité x doit tout d'abord calculer son accord avec les entrées qu'elle reçoit, cet accord étant son activation $i(x, t)$. Cet étage, que l'on pourrait appeler étage thalamique, monopolise une liste de liens, pour recevoir l'information, et un potentiel, pour héberger $i(x, t)$. Ce même potentiel

est aussi la sortie de cet étage. Si l'on en restait là pour la conception de l'unité, c'est cette sortie qui serait copiée dans A^* , le potentiel #0. Définissons maintenant l'étage de *compétition* par l'équation 3.6. Cet étage monopolise un potentiel, pour héberger $u(x, t)$, et deux listes de liens, une pour les liens latéraux excitateurs, et l'autre pour les liens latéraux inhibiteurs. Il faut fournir à cet étage la valeur d'entrée sur laquelle faire la compétition : ici il s'agit de la valeur $i(x, t)$ calculée par l'étage précédent. Cette conception par empilement d'étages est prise en charge par de la métaprogrammation C++ dans la bibliothèque que nous avons écrite, nous y reviendrons au paragraphe 9.6.3, mais retenons ici que cela permet de générer automatiquement, à partir du code, des figures en PDF qui sont cohérentes avec ce que l'on a programmé⁵⁸. C'est avec ces documents PDF que nous avons élaboré la figure 4.11 qui résume la construction de l'unité.

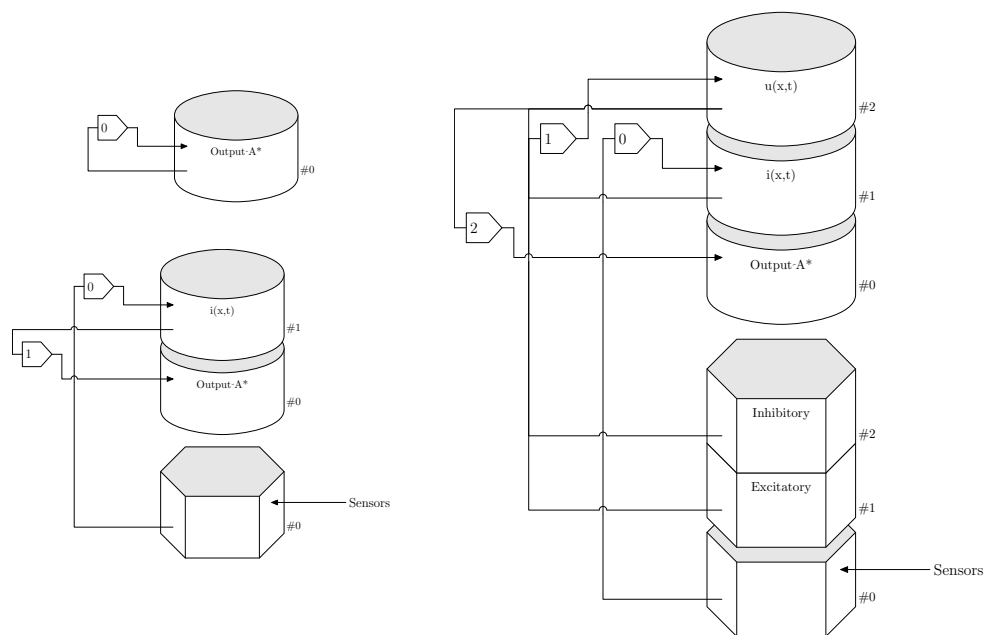


FIGURE 4.11 – Construction par empilement d'étages d'une unité réalisant l'équation d'un champ de neurones dynamiques. En haut à gauche, l'unité n'a qu'un étage, l'étage #0 qui est le seul autorisé en lecture. Dans cette unité dégénérée, la copie du résultat du calcul (rien) vers le potentiel #0 est l'opérateur 0 (les opérateurs sont représentés par des flèches étiquetées). En bas à gauche, l'étage « thalamique » est empilé (potentiel #1 et liste de liens #0). La façon dont $i(x, t)$ est calculé à partir de l'information reçue par les liens est l'opérateur 0, et la recopie de ce résultat vers le potentiel #0 est l'opérateur 1. À droite, l'étage de compétition est ajouté (potentiel #2, listes de liens #1 et #2). L'opérateur 1 est ici l'équation 3.6, et c'est l'opérateur 2 qui copie la sortie de l'étage de compétition vers A^* , le potentiel #0.

Ainsi, la conception d'un modèle en utilisant **bijama** revient à définir des étages, puis à les empiler, dans un ordre choisi, pour constituer des unités. Le même type d'étage peut-être utilisé à différentes niveaux d'un type d'unité à l'autre, voire être utilisé deux fois dans un même type d'unité.

⁵⁸. C'est un garde-fous extrêmement précieux lorsque le concepteur est confronté à la complexité du système qu'il programme.

4.5.2 Connexions en bandes et étages « *corticaux* »

Nous abordons dans ce paragraphe le cœur de la dynamique du modèle *bijama*, à savoir la connectivité en bandes. Cette connectivité et les calculs qui en résultent sont systématisés dans le modèle par un étage particulier, l'étage dit « *cortical* ». La notion de bandes dans le modèle est à rapprocher de ce que nous avons décrit pour le cortex au paragraphe 4.2.2.

Décrivons d'abord la topographie des connexions corticales du modèle. Dans le modèle, deux modules sont connectés via des connexions qui forment des faisceaux parallèles de liens, comme le précise la figure 4.12. L'orientation commune à tous ces liens est un paramètre de connexion, et elle doit être différente pour deux modules connectés à un même autre. Dit autrement, dans ce troisième module, les bandes de connexions doivent *se croiser*, cette exigence étant justifiée dans [Ménard, 2006]. Afin de ne pas favoriser une direction plutôt qu'une autre, les modules de *bijama* ont la forme d'un disque. Cette notion de bandes de connexions est plus claire si l'on considère le *champ récepteur* d'une unité particulière d'un module, au sens où nous l'avons défini au paragraphe 3.3. Si cette unité possède des liens provenant d'un autre module via une bande de connexion, ce champ, sur la surface de l'autre module, dessine une bande. Les champs récepteurs de toutes les unités du premier module dessinent des bandes parallèles, parfois confondues, parfois se recouvrant partiellement, sur la surface du deuxième module dont le premier reçoit l'information. Sur la figure 4.12-(A), outre les unités grisées qui montrent les champs récepteurs de l'unité du module du haut sur les deux autres, nous avons repéré deux unités du module supérieur par un point, plein pour l'une et ouvert pour l'autre. Pour l'unité marquée d'un point plein, nous avons marqué du même symbole les unités de son champ récepteur dans les deux autres modules. Nous avons procédé de même pour l'unité d'un point ouvert, mais dans le module en bas à droite, le champ récepteur se confond avec celui de l'unité marquée d'un point plein.

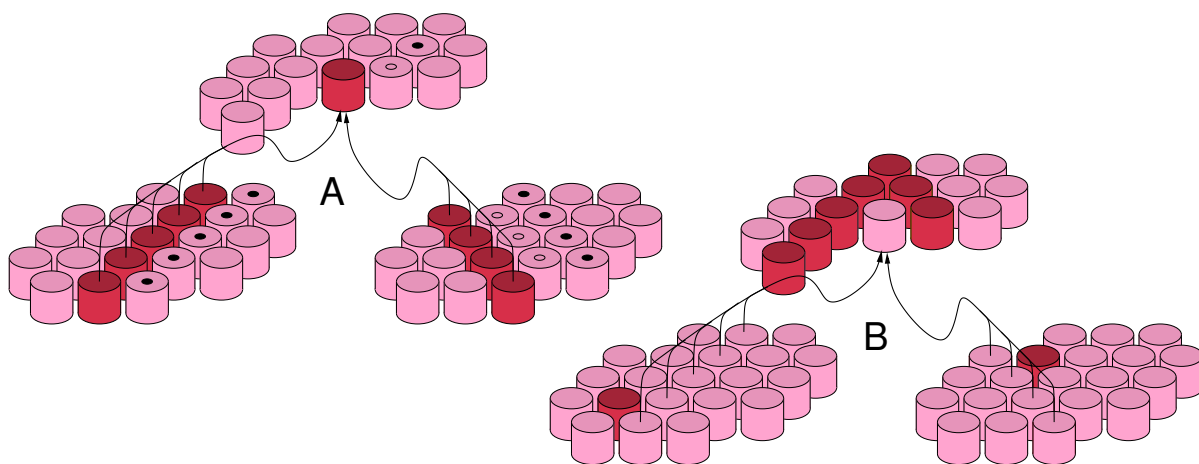


FIGURE 4.12 – Communication entre modules par des connexions en « *bandes* ». Les deux figures montrent la même chose, deux modules (en bas) connectés à un troisième (en haut). Pour l'unité plus sombre la carte du haut de la figure (A), les unités dont elle lit les activités via les connexions en bandes sont grisées dans les deux autres modules. Ces unités forment le champ récepteur de la première. Sur la figure (B), on voit dans le module du haut les champs récepteurs des unités grisées dans les deux modules du bas, dans le cas de connexions en bandes réciproques.

La connectivité en bandes dans *bijama* préside à la construction d'*indices primaires*, par une connectivité *topographique* (cf. page 80). Bien que du point de vue de la programmation, la

bibliothèque ne l'impose pas, dans le modèle, on construira de préférence des bandes réciproques. Cela signifie que si un module A lit un module B par des bandes, B lit également A par des bandes colinéaires, ce qu'illustre la figure 4.12-(B).

Un étage cortical est alors, dans ce modèle, la gestion d'une bande. Ce type d'étage monopolise un potentiel et une liste de liens. La liste de liens héberge les connexions vers la bande d'un module distant, et le potentiel est calculé en appliquant une opération sur les valeurs A^* des unités du champ récepteur, opération qui dépend également des poids des liens. Ainsi, sur la figure 4.12, les deux modules « *du bas* » ont des unités qui possèdent un seul étage cortical, pour recevoir les informations du module « *du haut* », alors que les unités du module du haut, elles, ont deux étages corticaux, un pour chaque module du bas.

4.5.3 Contraintes de cohérence multimodale

À partir de la notion de bandes de connexions, nous avons mis en place dans le modèle la notion d'*activation multimodale cohérente*. Ce type d'activation s'obtient en appliquant, au sein de modules liés par des bandes, une compétition par une équation de champs neuronaux dynamiques (cf. paragraphe 3.6.2). L'équation en elle-même est étudiée dans [Frezza-Buet and Ménard, 2005]. Illustrons ceci avec 4 modules. Les 3 premiers, que nous qualifierons de « *primaires* » sont connectés à un même quatrième module, « *associatif* », via des bandes qui sont orientées avec des angles différents, espacés de 120 degrés (cf. figure 4.14). Les modules primaires sont, dans cet exemple, soumis à une entrée, alors que le module associatif se limite à assurer la combinaison des trois modules primaires. La description des fonctions des modules est simplifiée si on la spécifie par un empilement d'étage.

Les 3 modules primaires ont des unités composées comme suit :

1. Un étage thalamique, calculant une activation $i(x, t)$ à partir d'une entrée extérieure.
2. Un étage cortical, calculant une activité d'après les potentiels A^* des unités du module associatif contenues dans le champ récepteur de l'unité primaires.
3. Un étage « *cortico-thalamique* » qui combine les potentiels des deux étages précédents.
4. Un étage de compétition.

Le module associatif, lui, a des unités composées des étages suivants :

1. 3 étages corticaux, un pour chacun des modules primaires.
2. un étage de « *synthèse corticale* », qui calcule un potentiel combinant les 3 potentiels des étages corticaux.
3. Un étage de compétition, *identique* à celui des modules primaires, bien que ce soit l'étage de synthèse corticale qui fournisse $i(x, t)$ pour la compétition.

La figure 4.13 décrit graphiquement ces deux types d'unités.

La dynamique de l'ensemble de l'architecture cherche à se stabiliser sur une position *cohérente*, ce qui est à rapprocher de la *restitution de cohérence* spatiale que nous avons mentionnée au paragraphe 3.6. La cohérence, ici, signifie qu'après compétition, les bulles d'activité dans les modules doivent se trouver à des positions *connectées*, alors que la connectivité par bandes est partielle (cf. figure 4.14). Le maintien d'activations entre les modules à des endroits connectés est une propriété de base de la multimodalité de *bijama*, nous y reviendrons. La dynamique de ce processus de relaxation complexe est illustrée sur la figure 4.15.

Sur la base de cet exemple, discutons de la notion de hiérarchie dans le modèle *bijama*. Le module associatif est là pour ajouter une contrainte aux modules primaires, contrainte de s'activer en des endroits dont les bandes se croisent *en un point* sur le module associatif. Réciproquement,

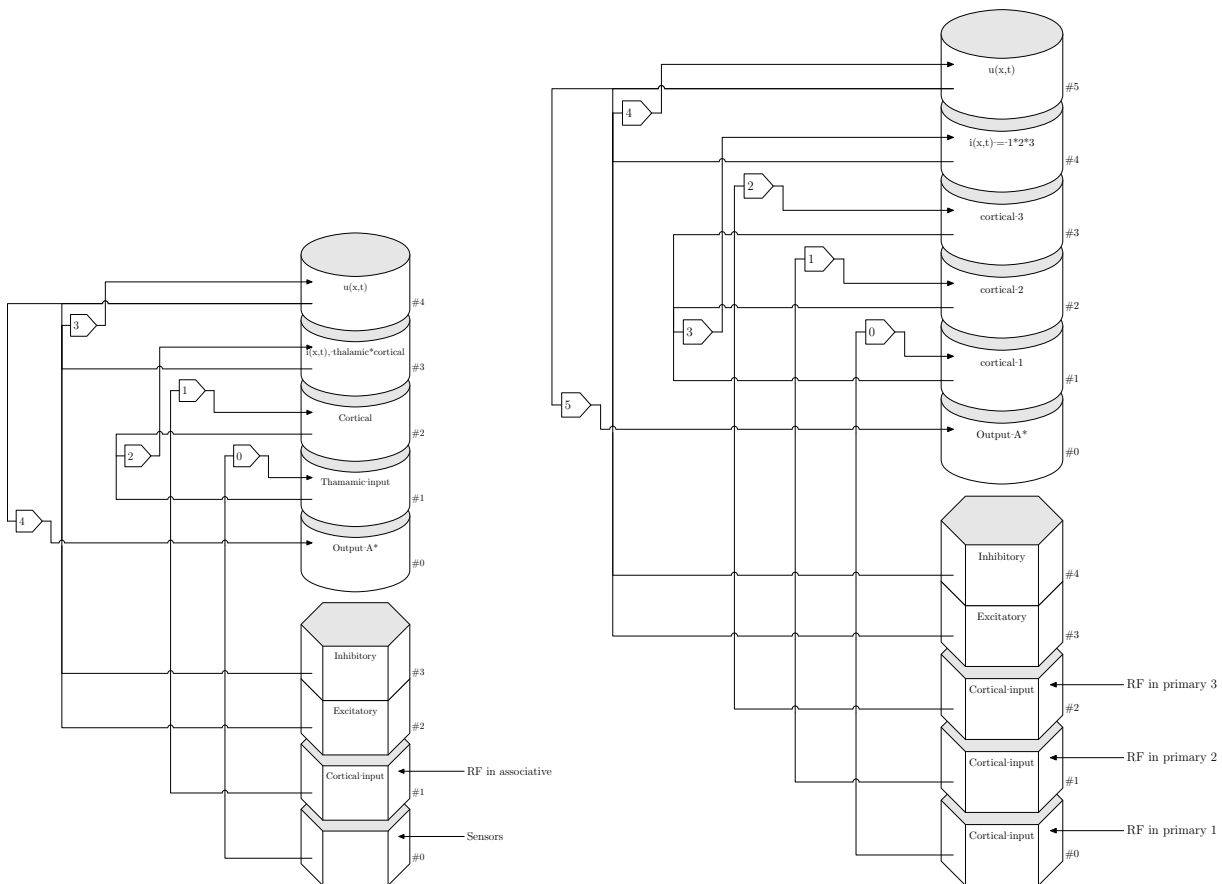


FIGURE 4.13 – À gauche est représenté le schéma d’une unité d’un module primaire. L’étage thalamique (potentiel #1, liens #0) est celui de la figure 4.11. L’étage cortical est le suivant (potentiel #2, liens #1). Les sorties de ces deux étages sont combinées par l’étage cortico-thalamique (potentiel #3), qui sert d’entrée à un étage de compétition analogue à celui de la figure 4.11. À droite est représenté le schéma d’une unité d’un module associatif. Les trois premiers étages sont des étages corticaux, analogues à celui de la figure de gauche. Les sorties de ces trois étages sont combinées par l’étage de synthèse corticale (potentiel #4), qui sert d’entrée à un étage de compétition.

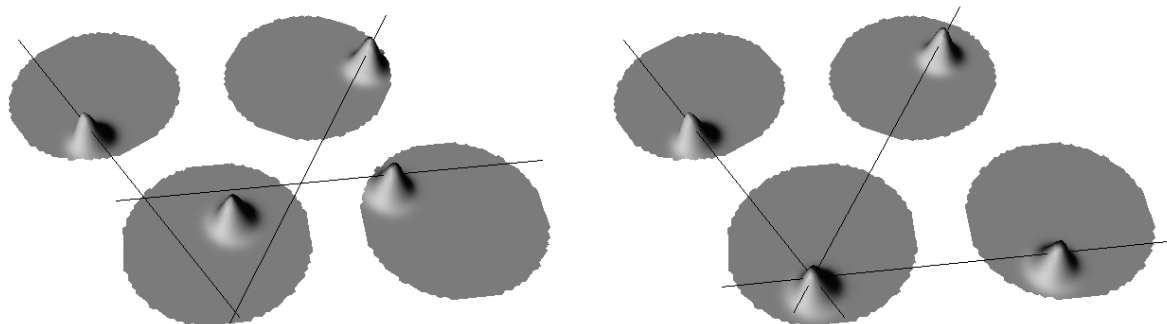


FIGURE 4.14 – Relaxation vers un état de cohérence entre modules. Le module associatif parmi les 4 est celui du bas, à gauche, les autres étant les modules primaires. Le potentiel de sortie A^* (copie du potentiel #4 pour les modules primaires, du potentiel #5 pour le module associatif, cf. figure 4.13) est représenté sur les champs. À gauche, l'état est incohérent car les bulles ne se situent pas à des endroits connectés entre eux par les bandes inter-modules. C'est un état instable. À droite au contraire, l'état est un état cohérent et sable : les bulles se situent à des endroits effectivement reliés.

ce module capte l'état des modules primaires pour déterminer cet endroit de croisement. Ainsi, les connexions en bandes permettent une relaxation, une résonance⁵⁹. Les relations d'influences mutuelles entre les modules, dans ce modèle, sont alors beaucoup plus symétriques que celles que nous avons décrites au paragraphe 4.2.3. C'est ainsi que notre modèle remet en cause la notion de hiérarchie, comme nous l'avons discuté au paragraphe 4.2.4 et plus généralement au paragraphe 1.3.3.

4.5.4 Principe d'*auto-organisation*

L'architecture *bijama* met au premier plan la notion d'apprentissage compétitif, la compétition étant réalisée une équation de champ neuronal (cf. 3.6.2). Ainsi, chaque module (i.e. chaque carte) est le siège d'un apprentissage analogue à ce que Kohonen décrit dans [Kohonen, 1997], dont il a tiré par la suite les cartes auto-organisatrices. Un module peut donc se ramener au schéma de la figure 4.16. L'étage supérieur est l'étage de compétition, calculant la distribution d'activité u du module. Il s'agit, comme nous l'avons dit, d'un champ neuronal. L'étage inférieur, qui fournit au champ neuronal la distribution des entrées i , représente l'empilement des étages qui calculent une activité de reconnaissance i soumise à compétition. Dans l'exemple de la figure, nous considérons le cas où cette reconnaissance est la comparaison d'une entrée ξ à un prototype ω . Il s'agit alors d'un étage que nous avons qualifié de « thalamique » au paragraphe 4.5.1. En permanence, l'étage thalamique de chaque unité adapte son prototype ω de sorte qu'il se rapproche de l'entrée courante. Toutefois, cette activité permanente est modulée par la sortie de l'étage de compétition, ce qui fait qu'en pratique, seules les unités dont l'activité u est non nulle réalisent effectivement cet apprentissage.

Dans [Alecú and Frezza-Buet, 2008; Alecú *et al.*, 2011], nous avons appliqué cette archi-

59. Les règles de combinaison d'activations au niveau des étages respectent les principes de résonance introduits par Grossberg, voir [Ménard and Frezza-Buet, 2005].



FIGURE 4.15 – Dynamique de restitution de cohérence dans le modèle *bi_jama*. Le module central est le module associatif. La surface pleine grise correspond au potentiel de sortie A^* (copie du potentiel #4 pour les modules primaires, du potentiel #5 pour le module associatif, cf. figure 4.13). Les surfaces « *en grille* » sont le potentiel jouant le rôle d'entrée dans la compétition (potentiel #3 pour les modules primaires, #4 pour le module associatif). Dans le sens de lecture : la première image représente un état stabilisé en l'absence d'entrées. On impose alors au système une entrée, maintenue, sur un des modules, de sorte qu'une zone localisée réponde par une activation corticale (potentiel #1) à un endroit précis, marqué d'une flèche sur la première image. Les images 2 et 3 montrent l'évolution de l'ensemble du système, qui se stabilise dans un nouvel état cohérent. Dans un autre module, on impose alors une entrée localisée et maintenue, comme précédemment, à l'endroit marqué d'une flèche sur la quatrième image. Les images suivantes montrent les étapes qui conduisent à la nouvelle stabilisation du système. On notera, au cours de cette stabilisation, le glissement de la bulle dans le module associatif, et le glissement du profil de potentiels dans le module de droite.

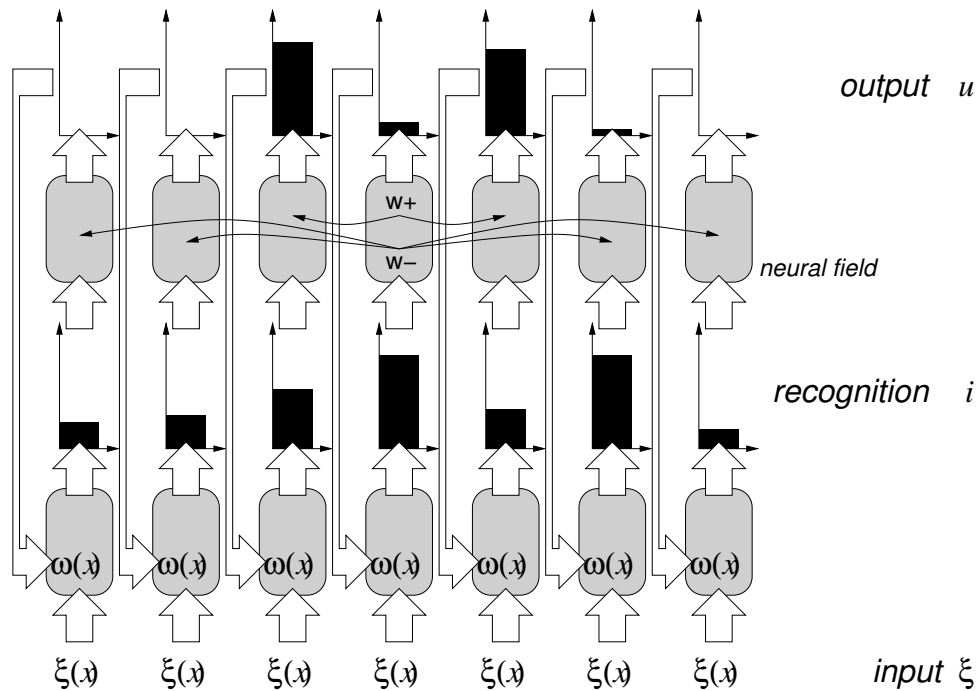


FIGURE 4.16 – Architecture distribuée réalisant le principe d’auto-organisation. D’après [Alecu *et al.*, 2011]

tecture à quelques problèmes d’auto-organisation simples et montré qu’il n’est pas immédiat d’obtenir un effet analogue aux cartes auto-organisatrices de Kohonen sur la base de cette architecture. En effet, si l’on s’astreint à des calculs strictement locaux, si l’on s’interdit de remettre artificiellement à zéro l’activité u lorsqu’une nouvelle entrée est présentée, et si on tolère une initialisation aléatoire des prototypes ω , la dynamique de l’ensemble du module ne conduit pas à une auto-organisation avec les équations usuelles de champ neuronal. C’est une difficulté qu’à rencontrée Olivier Ménard au cours de sa thèse [Ménard, 2006], et qui nous a conduit à concevoir une nouvelle équation de champ neuronal décrite au paragraphe 4.5.6. Le caractère « *pernicieux* » de la dynamique d’auto-organisation sous ces conditions est décrit dans [Alecu *et al.*, 2011]. Il tient en partie au fait que cette dynamique change au fur et à mesure de l’élaboration de l’auto-organisation, comme l’illustre la figure 4.17.

Nous avons constaté empiriquement dans [Alecu *et al.*, 2011] que le critère de qualification des champs neuronaux dynamiques proposé par Lucian Alecu (cf. paragraphe 3.6.3) permet de savoir si une équation de champ neuronal peut réaliser l’auto-organisation de la structure de la figure 4.16.

4.5.5 Principe de l’*auto-organisation conjointe*

Nous avons réalisé deux types d’apprentissage dans le modèle *bijama*. Le premier se situe au niveau de l’étage thalamique, et correspond à l’ajustement des poids pour modifier la configuration des entrées à laquelle l’unité est la plus sensible. En s’éloignant un peu de la biologie, on peut remplacer cet étage par un apprentissage comme celui opéré dans les algorithmes de quantification vectorielle (modification du prototype), nous fournissons d’ailleurs un étage pour cela dans la bibliothèque. C’est ce qui est décrit au paragraphe précédent. Le second type d’apprentissage s’effectue au niveau des poids des liens qui constituent les bandes de connexion inter-modules.

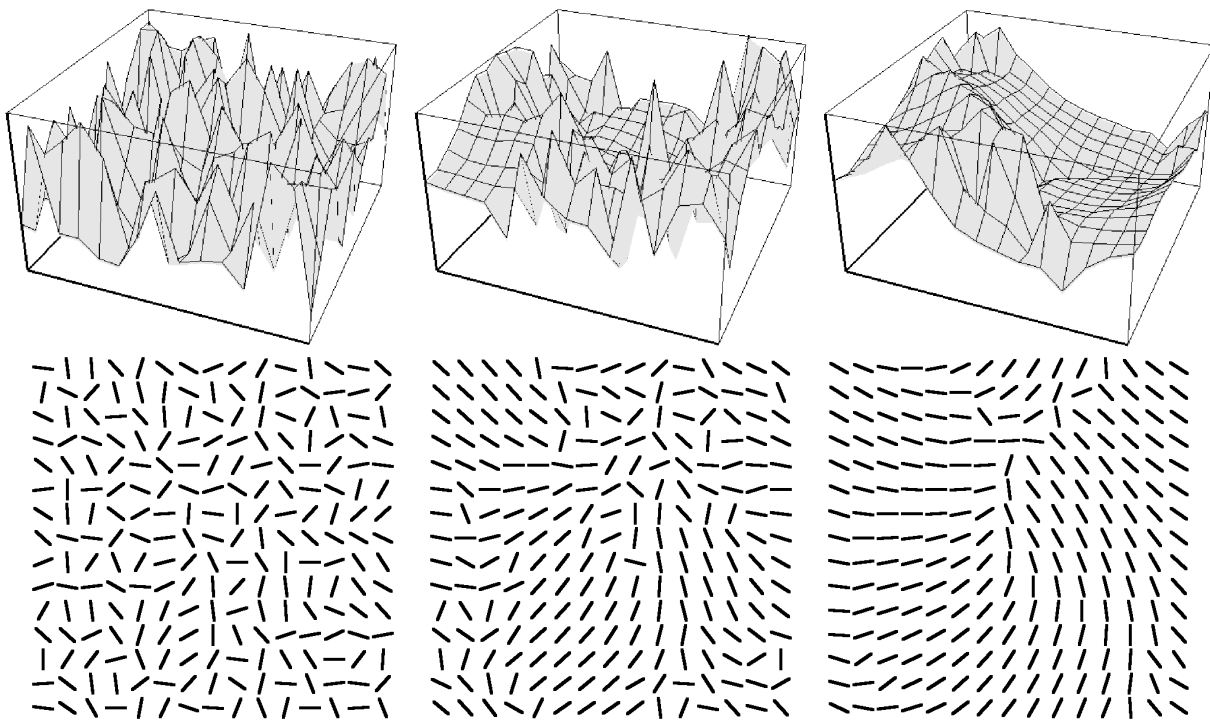


FIGURE 4.17 – En haut, de gauche à droite, évolution de la distribution i au fur et à mesure de l'établissement au sein de la carte d'une auto-organisation. L'exemple d'auto-organisation, en bas, est celui d'une orientation de contraste (ω est une orientation). L'architecture utilisée est celle de la figure 4.16. D'après [Alecu *et al.*, 2011]

C'est sur ce point que nous allons nous attarder.

*Dans le modèle **bijama**, tous les apprentissages ont lieu en permanence, à chaque mise à jour d'une unité, et ces apprentissages tolèrent une initialisation aléatoire des poids. Ce qui varie, c'est le taux d'apprentissage, modulé par l'activité globale A^* de l'unité. Ainsi, alors que la règle est effectivement appliquée systématiquement, seules les unités d'un module qui se situent dans une bulle d'activité à un instant donné apprennent, cet apprentissage portant simultanément sur leur étage thalamique, quand elles en ont un, et sur le ou les étages corticaux qui la composent.*

L'activation du (ou des) potentiel(s) d'un étage a toujours, dans *bijama*, une sémantique booléenne. Ainsi l'étage thalamique dit si, oui ou non, l'entrée est dans la configuration préférée de l'unité. Le degré d'activation de l'étage est alors une information floue, au sens qu'a ce terme en logique floue⁶⁰. Les combinaisons entre étages, réalisées par d'autres étages, correspondent à la mise en œuvre d'opérateurs booléens flous, qui sont des opérateurs **et** pour ceux que nous avons décrits⁶¹.

*Nous affirmons avec le modèle **bijama** que soutenir l'idée d'un codage tabulaire (cf. page 4.2.1) dans un modèle cortical promeut l'affirmation d'une manipulation, floue, d'informations booléennes par les unités de ce modèle.*

Ce point de vue a un impact sur la nature des règles d'apprentissage des poids d'une bande de connexion, et sur la règle d'activation du potentiel de l'étage cortical correspondant, ce qui est discuté dans [Ménard, 2006]. Nous en retiendrons ici que l'étage cortical d'une unité apprend à ne s'activer que pour l'activation de certaines unités du champ récepteur, ce qui a pour effet, quand l'apprentissage se stabilise, de focaliser plus systématiquement la dynamique vers les états stables appris.

Afin de cerner l'effet qui émerge de cette dynamique d'apprentissage, considérons le cas d'étude que nous avons abordé durant le projet MirrorBot (cf. paragraphe 8.1.3), concernant la modélisation de l'influence de la sémantique sur l'organisation de cartes phonétiques [Pulvermüller, 2003]. L'architecture *bijama* construite pour cette expérimentation est celle illustrée sur la figure 4.18.

On soumet au modèle des mots, sous forme de vecteurs acoustiques, en activant les unités relatives à l'action correspondante dans le module dédié aux parties du corps impliquées par les actions (« *Action representation* »). On s'intéresse à l'auto-organisation qui s'établit dans le module acoustique, noté « *Word representation* » sur la figure 4.18.

Une première expérience consiste à isoler le module acoustique, en supprimant son étage cortical. Il se comporte alors comme une carte auto-organisatrice. La figure 4.19 montre le résultat de l'organisation pour deux expériences. Ces résultats montrent qu'une conservation de topologie se produit : deux unités proches sur la carte détectent préférentiellement des mots proches d'un point de vue acoustique. Ceci correspond à l'établissement d'*indices secondaires* (cf. page 4.2.1), le module pouvant se ramener ici à une maxicolonne (cf. page 4).

60. Notre modèle ne partage cependant pas le caractère symbolique et inférentiel de la logique floue.

61. Le **et** est représenté par le signe * sur la figure 4.13.

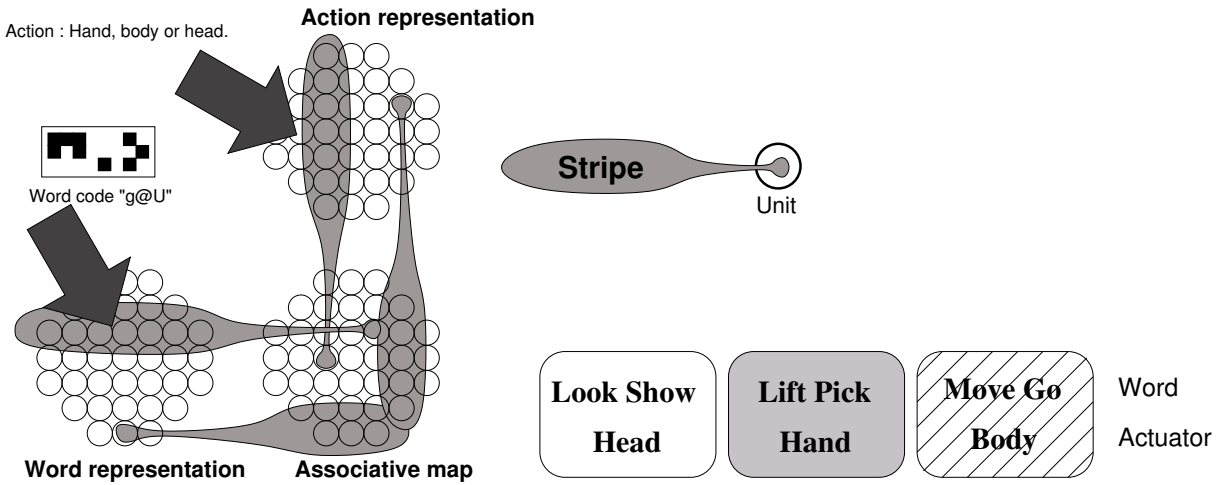


FIGURE 4.18 – L’architecture se compose de deux modules primaires et d’un module associatif. L’un des modules primaires (en haut) reçoit comme entrée thalamique la modalité somesthésique, très rudimentaire dans ce modèle puisque l’on coupe le module en trois bandes verticales. L’utilisation de la tête active l’étage thalamique des unités de la bande de gauche, celle du corps la bande du milieu, et celle de la main la bande de droite. Les unités de l’autre module primaire (à gauche) reçoivent toutes un même vecteur de primitives acoustiques correspondant au mot prononcé. Un code de couleur (blanc, rayé, gris) est défini pour représenter la sémantique des mots en relation avec une partie du corps (en bas à droite de la figure).

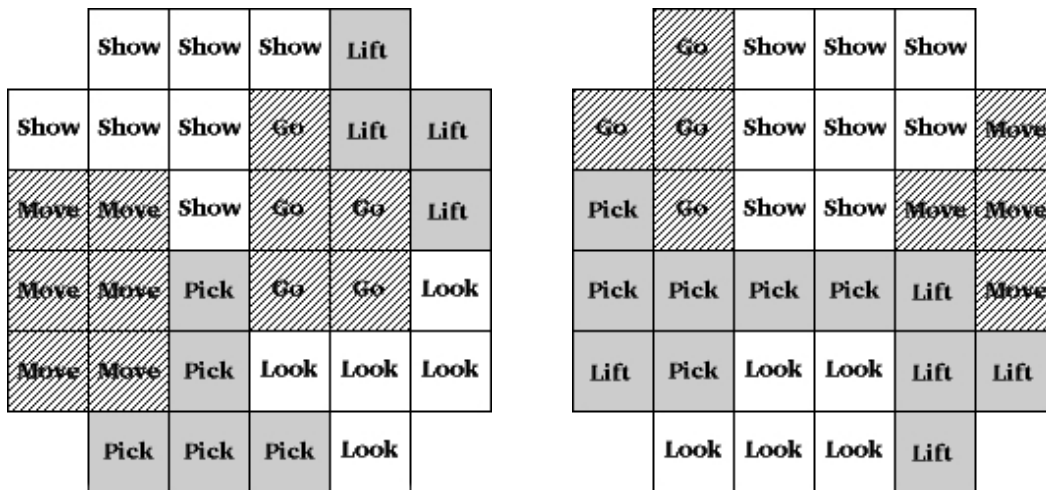


FIGURE 4.19 – Résultat de l’auto-organisation pour deux expériences sur un module acoustique isolé du reste de l’architecture. On représente à l’endroit de chaque unité le mot pour lequel son étage thalamique répond le plus. La couleur rappelle la partie du corps associée, selon la légende de la figure 4.18.

Lorsqu'on insère le module acoustique dans l'architecture, le système de cohérence inter-cartes influence le module acoustique par le module relatif au corps, par le truchement du module associatif. Il en résulte l'auto-organisation du module acoustique présentée sur la figure 4.20 pour deux réalisations différentes de l'expérience.

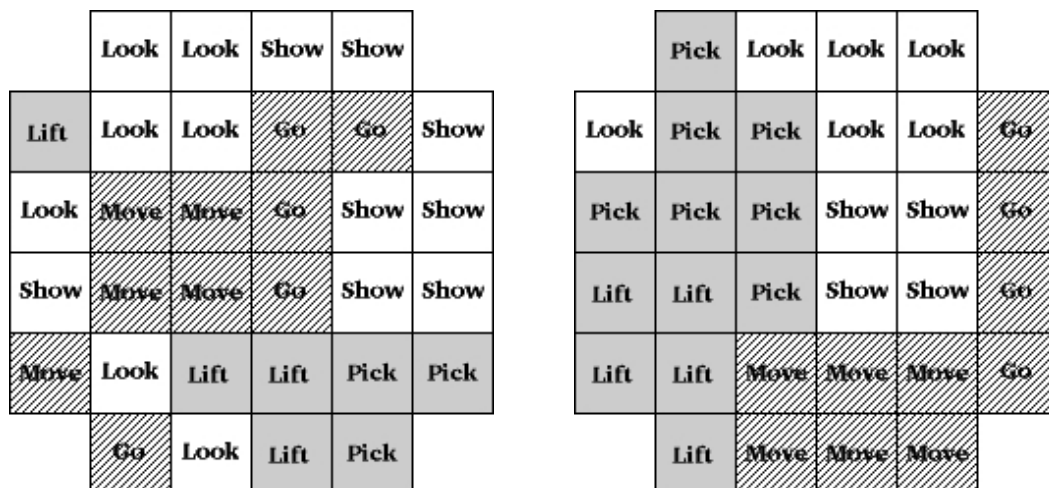


FIGURE 4.20 – Résultat de l'auto-organisation pour deux expériences sur un module acoustique connecté au reste de l'architecture, avec des conventions graphiques identiques à celles de la figure 4.19.

On constate alors que l'organisation est toujours acoustique, car les mots proches sur la surface du module sont des mots proches d'un point de vue acoustique, mais que maintenant, les mots sont aussi proches d'un point de vue sémantique, les couleurs associées aux parties du corps formant des régions non morcelées.

Sans remettre en cause l'auto-organisation basée sur les caractéristiques de l'entrée thalamique, l'architecture bijama profite du fait qu'il existe différentes auto-organisations possibles pour un module si l'on ne devait tenir compte que l'entrée thalamique. L'existence de ce degré de liberté est exploitée implicitement pour que le module converge vers une auto-organisation qui respecte des contraintes globales, comme la contrainte de sémantique corporelle de notre exemple.

On voit ici apparaître que des contraintes de cohérences globales s'expriment en chaque module, du fait du maintien d'activations cohérentes entre modules par le processus de relaxation conjoint entre les cartes. À notre connaissance, cette propriété n'a pas été exhibée par d'autres modèles informatiques.

4.5.6 Le champ de neurone BINP

Nous avons soulevé au paragraphe 4.5.4 la difficulté d'utilisation d'un champ neuronal pour mettre en œuvre la compétition au sein d'un module auto-organisant [Alecu *et al.*, 2011]. Cette difficulté nous a conduit à proposer, dans le cadre de la thèse de Lucian Alecu, une nouvelle équation d'évolution du potentiel de sortie u d'un champ neuronal [Alecu and Frezza-Buet, 2009a; Alecu *et al.*, 2011].

Ce champ, à l’instar du modèle proposé dans [Pinto and Ermentrout, 2001a; Pinto and Ermentrout, 2001b], repose sur l’utilisation de deux potentiels u et v . u correspond classiquement à l’établissement d’une bulle d’activité, mais v est une bulle plus large, qui englobe u , et dont la croissance survient avec un délai. Si le champ ne reçoit que des entrées faibles, le terme v est inhibiteur (*back inhibition*), de sorte à reconsidérer la position d’une bulle qui aurait émergé au sein d’une distribution i faible. Si en revanche la bulle se forme à l’endroit d’une forte activité i alors l’inhibition exercée par v est inopérante, et le champ se comporte classiquement. Ce procédé, illustré sur la figure 4.22 est essentiel pour assurer l’auto-organisation, comme justifié dans [Alecu et al., 2011]. Il est assuré par le terme g de l’équation 4.1, terme représenté sur la figure 4.21.

$$\begin{aligned} \frac{1}{\tau} \cdot \frac{du(x, t)}{dt} &= i(x, t) + \alpha \cdot \mathcal{E}(x, t) - \beta \cdot \mathcal{I}(x, t) - \gamma \cdot g(i, v) & (4.1) \\ \frac{dv(x, t)}{dt} &= h(\mathcal{E}(x, t)) \\ \mathcal{E}(x, t) &= \sum_{x'} w^+(\|x - x'\|) f(u(x', t)) \\ \mathcal{I}(x, t) &= \sum_{x'} w^-(\|x - x'\|) f(u(x', t)) \end{aligned}$$

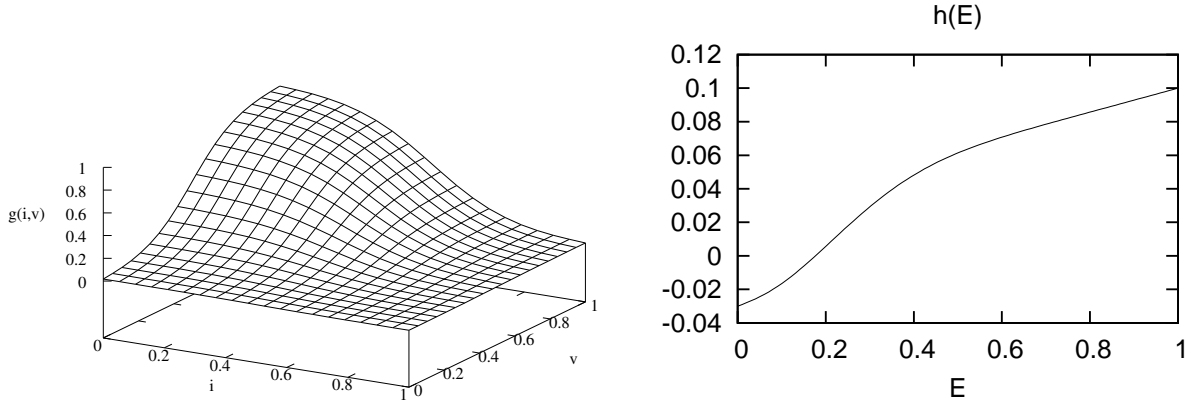


FIGURE 4.21 – Termes $g(i, v)$ et $h(\mathcal{E})$ apparaissant dans l’équation 4.1.

Le champ de neurone BINP (*Back Inhibited Neural Population*) que nous proposons est le principe de compétition que nous proposons par défaut dans l’architecture `bijama`.

4.5.7 Extensions

Le modèle `bijama`, aujourd’hui, fournit des étages prêts à l’emploi pour concevoir les unités d’une architecture. Les principes d’auto-organisation influencée localement par des considérations globales à l’architecture sont bien établis par le modèle. Toutefois, les opérateurs de combinaison des étages, ainsi que l’équation de compétition ne sont pas encore stabilisés. Ces derniers points font l’objet de nos recherches actuelles.

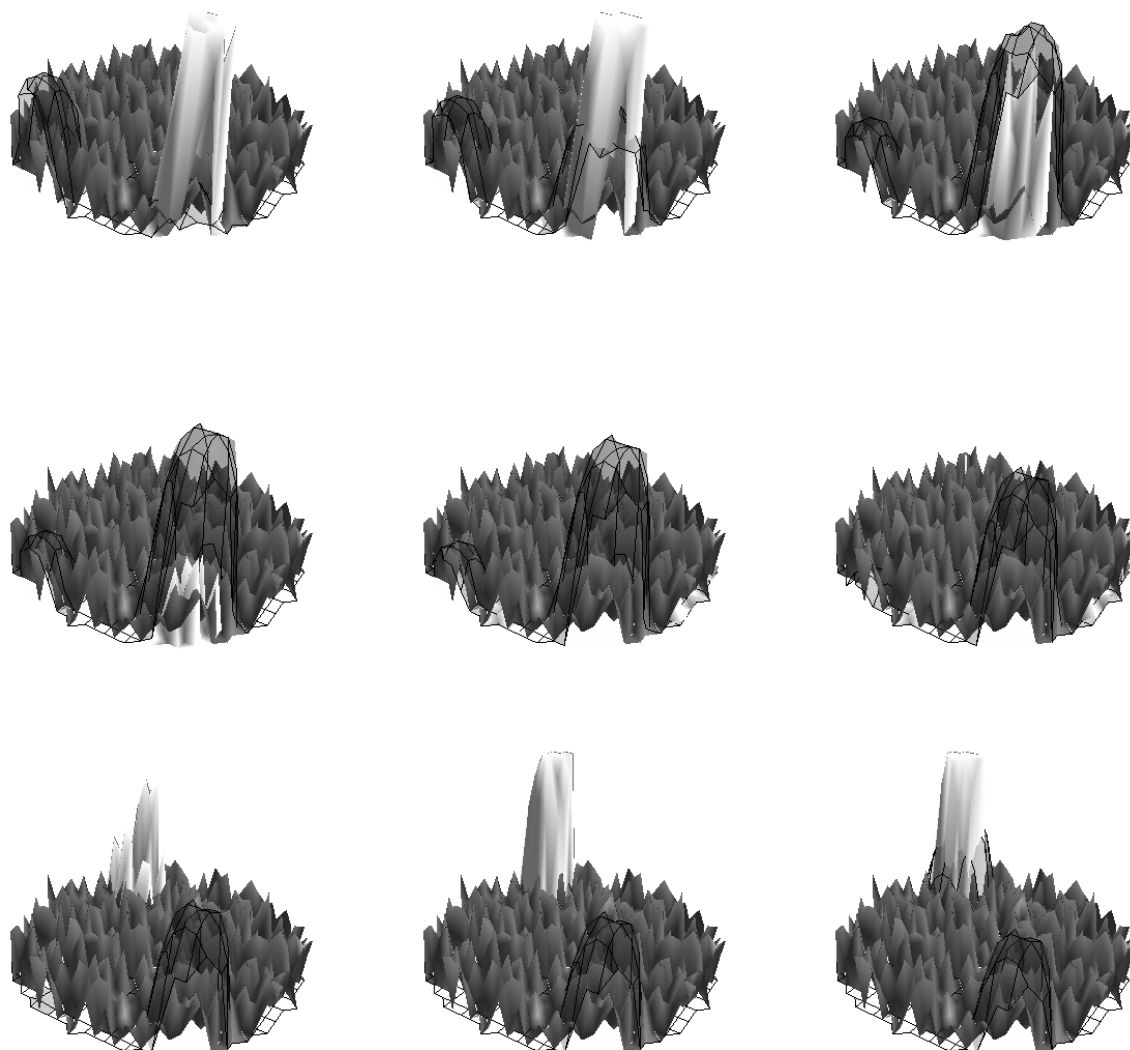


FIGURE 4.22 – La formation d’une bulle de potentiel u (en clair) en réponse à une entrée i faible (en sombre) est suivie de la montée du potentiel v (en grille). Ce potentiel inhibe u de sorte que la bulle puisse se reformer ailleurs.

Chapitre 5

Approche située et calcul cortical

Dans le chapitre précédent, nous avons présenté le cortex comme un modèle, fourni par la nature, pour concevoir des architectures de calcul propres à faire progresser l'informatique, en contestant dans la perspective de nos objectifs les piliers de notre discipline que sont les notions de modularité, d'interface, de séquentialité.

Il est clair qu'il ne s'agit pas ici de généraliser cette contestation à l'ensemble de ce qui est abordé par l'informatique aujourd'hui, mais plutôt de considérer cette contestation là où elle porte effectivement, à savoir la conception de systèmes d'information propres à doter un système autonome (un robot typiquement) d'une intelligence située. C'est ce que nous avons articulé dans l'ensemble du chapitre 2.

Dans l'état actuel de nos travaux, à quelques exceptions près toutefois, nous n'avons pas encore fait la preuve expérimentale qu'un calcul cortical, au sens où nous l'avons défini au chapitre 4, permet de produire une intelligence conduisant à des raisonnements articulés qui soient également situés, couplés avec l'environnement dans lequel évolue l'agent à qui cette intelligence est conférée. C'est pourtant ces points-là, introduits au chapitre 1, que nous souhaitons aborder par nos modèles.

La promotion du calcul cortical, que nous avons instancié par le modèle *bijama*, est la promotion de principes informatiques comme l'auto-organisation conjointe, la décision collective, le recrutement de ressources, la prise en charge de la multimodalité, etc. On peut dire de façon équivalente que les approches symboliques de l'intelligence artificielle font elles aussi la promotion de principes élégants, comme le raisonnement à partir de cas, l'inférence logique, les résolution de problèmes de détour, etc. Dans l'état actuel de nos travaux, nous n'avons pas plus que les tenants des approches symboliques fait la preuve de la possibilité de nos systèmes à ancrer une intelligence dans le monde. D'autres chercheurs, comme par exemple ceux du laboratoire ETIS dirigé par Philippe Gaussier ont, eux, clairement établi cette preuve pour leurs modèles, puisque ces modèles s'instancient d'emblée sur des robots. On peut bien sûr dire la même chose de la *subsumption architecture* de Brooks.

Il nous faut donc, pour confirmer l'adéquation du calcul cortical avec les objectifs présentés au chapitre 2, montrer qu'un modèle cortical, comme *bijama*, est un pas vers des développements qui conduisent d'une part à l'intelligence, et d'autre part à la *situation* des systèmes. C'est tout l'objet de ce chapitre.

Nous avons donné en introduction à ce mémoire, au paragraphe 1.3.2, quelques éléments de réponse à cette question, en soutenant l'idée que la cognition et la perception sont des concepts fondamentalement moteurs. Ce que nous nous proposons de discuter ici, plus précisément que nous ne l'avons fait en introduction, est la façon dont cette affirmation, générale, se décline

dans le cas particulier du calcul cortical. Cette vision centrée sur la motricité a l'avantage de concilier les deux voies de développement que nous avons mentionnées dans notre problématique au chapitre 2, puisque d'une part la motricité de la perception et des actions est indissociable du comportement, ce qui répond à l'ambition de situer le calcul cortical, et puisque d'autre part la motricité est selon nous la voie vers l'intelligence, animale comme artificielle.

5.1 Le cortex réalise une cognition motrice

Le fait que la cognition soit motrice est soutenu pour l'ensemble du cerveau, et n'est certes pas propre au cortex, comme le souligne Alain Berthoz que nous avons cité page 16. Dans le cortex toutefois, à l'instar de ce qui se passe dans les autres grandes structures homogènes du système nerveux (cervelet, hippocampe, striatum, etc.), cette propriété est instanciée sur un support de traitement « *intéressant pour l'informatique* », comme nous l'avons défendu, mais surtout elle se reflète dans la prise en charge de la cognition, dont un des « *fleurons* » est le langage.

La biologie du cortex aborde aujourd'hui la question de l'engrammation sur le substrat cortical de représentations conceptuelles, ce dont nous rendons compte ici, trop succinctement, par les travaux de Victorio Gallese et Friedmann Pulvermüller dont nous avons eu connaissance par notre participation au projet *MirrorBot* (cf. paragraphe 8.1.3).

Le cortex apporte fondamentalement une contestation du caractère abstrait, voire éthéré, de ce que nous appelons des concepts. Alors que notre raisonnement nous paraît « *en retrait* » de notre incarnation dans le monde, ce qui a motivé la promotion des approches symboliques de l'intelligence, la biologie du cortex nous enseigne que ce retrait que nous percevons n'est pas étayé par l'organisation sur le substrat cortical de nos compétences pour l'abstraction et le langage. Nous renvoyons le lecteur à [Gallese and Lakoff, 2005] pour une discussion plus argumentée.

« First-generation cognitive science was strongly influenced by the analytic tradition of philosophy of language, from which it inherited the propensity to analyse concepts on the basis of formal abstract models, totally unrelated to the life of the body, and of the brain regions governing the body's functioning in the world » .

Victorio Gallese et George Lakoff [Gallese and Lakoff, 2005]

Ce que l'on apprend au contraire, c'est que le traitement des aspects les plus abstraits de notre cognition sont gérés par des structures sensorimotrices, « *recyclées* » pour le raisonnement et le langage.

« [...] a key aspect of human cognition is neural exploitation – the adaptation of sensory-motor brain mechanisms to serve new roles in reason and language, while retaining their original functions as well » .

Victorio Gallese et George Lakoff [Gallese and Lakoff, 2005]

Ce qui est important, comme le soulignent les auteurs, c'est qu'il n'y a pas de centre dédié à la manipulation symbolique abstraite, mais que cette manipulation est répartie au niveau des diverses structures sensorimotrices. Mais là où cette remarque est particulièrement intéressante pour nous, c'est quand les auteurs soutiennent que les zones ainsi « *recyclées* » sont des zones qui *conservent* leur fonction initiale. Il devient, suivant cette perspective, pertinent d'étudier les primates non humains pour étudier le langage, puisque ce dernier repose sur des compétences que nous partageons avec eux. Les auteurs ont réalisé des expériences qui soutiennent l'hypothèse qu'imaginer quelque chose, c'est le simuler, au moyen des structures qui auraient pris en charge

cette chose si elle était survenue effectivement. Or, selon eux (et nous adhérons à ce point de vue), comprendre quelque chose, c'est être capable d'invoquer cette simulation, d'imaginer. Ayant en tête cette définition de la compréhension comme capacité à imaginer, les mots de Gallese et Lakoff nous permettent d'aborder la question de la sémantique.

« [...] understanding is imagination [...] what you understand of a sentence in a context is the meaning of that sentence in that context » .

Victorio Gallese et George Lakoff [Gallese and Lakoff, 2005]

Nous nous appuyons sur ces auteurs pour soutenir que le lien entre la structure symbolique d'une phrase et sa sémantique tient, dans le cortex, à ce que la prise en charge de la structure de la phrase est faite sur le tissu cortical au même endroit que le *comportement* auquel cette phrase réfère. Cette propriété est très nettement mise en avant par des études à l'IRMf [Hauk et al., 2004] qui montrent la relation entre la répartition du codage des mots d'action sur le substrat cortical et celle des représentations du corps.

« [...] the sensory-motor system not only provides structure to conceptual content, but also characterizes the semantic content of concepts in terms of the way that we function with our bodies in the world. » .

Victorio Gallese et George Lakoff [Gallese and Lakoff, 2005]

Ainsi, nous refusons dans notre approche de la cognition située de séparer la forme du langage, le caractère prédicatif et symbolique des raisonnements, de la sémantique, soutenant que la cohérence entre forme et sens, c'est-à-dire le caractère situé du langage et des raisonnements, tient à ce que ils s'inscrivent dans le corps et la sensorimotricité, si l'on en croit la structure du cortex.

Ce point de vue fait d'ailleurs écho au caractère moteur du langage, tel qu'il est soutenu par Fuster, comme nous l'avons résumé au paragraphe 1.3.2.

Nous pensons donc que la clé de la cognition tient à ce que le cerveau est un organe moteur, et qu'au niveau de la structure générique du cortex, cette propriété motrice s'exprime sous la forme de raisonnements, de langages, s'appliquant à des calculs qui s'éloignent du contrôle direct des muscles, sans en perdre les caractéristiques. Nous nous trouvons a priori, du fait de notre investigation des fonctionnalités sensorimotrices corticales que nous avons opérationnalisées par le modèle *bijama*, dans la situation où « *Il n'y a qu'à* » appliquer le modèle à des robots, et voir jusqu'où nous serons capables de généraliser ces fonctions sensorimotrices vers la notion de raisonnement articulé.

Malheureusement, les choses ne sont pas si immédiates, et « *il n'y a* » pas « *qu'à* » . La première limitation est celle de la lourdeur logistique d'une plate-forme robotique et de la quantité de logiciels associés, que nous discutons au paragraphe ???. Cette limitation n'est toutefois que logistique, contrairement à une seconde limitation, plus épineuse, articulée dans ce qui suit.

Les modèles sur lesquels nous avons travaillé depuis que nous sommes à Supélec sont des modèles associatifs, contrairement à ceux que nous avons développés lors de nos travaux de thèse, où la question de séquences comportementales était bien plus centrale. Nous avons laissé de côté ces derniers car nous percevions la nécessité de travailler plus finement le codage distribué de l'information et l'auto-organisation multimodale, ce qui a conduit à la définition de *bijama*, présenté au chapitre 4.

Il n'en reste pas moins vrai qu'en l'état actuel des choses, situer le calcul cortical suppose de dépasser le cadre de la « simple » association multimodale que nous avons étudiée pour, à nouveau, réintroduire la marque des exigences comportementales dans les modules corticaux, et ainsi viser l'émergence de raisonnement prédictif situé.

Nous avons, lors de la thèse d'Olivier Ménard, fait un pas dans cette direction, ce que nous présentons au paragraphe 5.4. Bien entendu, en cohérence avec la démarche présentée dans l'ensemble de ce document, il n'est pas question d'introduire ces exigences comportementales en les explicitant au niveau des unités de nos futurs modèles. Nous préférons faire porter cette évolution nécessaire de notre modèle sur les mécanismes qui puissent assurer une émergence, au sein de nos modules, de la prise en compte de l'exigence comportementale.

Au niveau des mécanismes, il nous apparaît que les deux notions qui manquent à notre version actuelle du calcul cortical pour aborder la cognition située sont d'une part la prise en compte de la notion récompense, et d'autre part une gestion du temps. Nous voyons ces points comme les pendants, au niveau microscopique, des propriétés macroscopiques que nous attendons des versions futures de nos algorithmes.

5.2 Les temps

Dire qu'il manque la prise en compte *du* temps dans notre approche actuelle du calcul cortical est mal formulé, dans la mesure où, même en restant dans le cadre informatique qui est le nôtre et sans aborder la question d'un point de vue philosophique, il convient, pour articuler cette notion, de la mettre au pluriel.

Nous parlons donc, dans ce paragraphe, *des* temps, et nous abordons la question par le constat des différentes échelles de temps en jeu dans les modèles, les mécanismes neuronaux [Frezza-Buet *et al.*, 2001], ainsi que dans les observations des transmissions de signaux dans le cerveau, comme avancé pour la vision dans [Hupe *et al.*, 1998] par exemple. Il est avéré que cohabitent dans le cerveau des processus rapides avec des plus lents, et ce même au sein d'un même module, comme le proposent les modèles à base de traces qui ont des durées d'extinction variables, plusieurs traces de constantes de temps différentes existant pour un même événement (voir [Grossberg and Schmajuk, 1989; Dominey, 1995] par exemple).

Il nous apparaît toutefois que catégoriser les mécanismes temporels suivant leur constante de temps n'est pas ce qui est le plus pertinent, et que les différentes constantes de temps sont des reflets d'une autre typologie des traitements temporels. Nous proposons ainsi de considérer trois types de temps dans le cerveau, ou tout du moins dans les systèmes informatiques que l'on peut en dériver.

Le premier est ce que nous appellerons le *temps physique*, c'est celui qui se mesure en secondes. Le mécanisme qui lui est associé est la *trace*, qui est une fonction qui est maximale juste après la survenue d'un événement (ou en même temps), puis qui décroît ensuite. Appliquer des apprentissages associatifs sur les traces permet de capturer des corrélations temporelles sur la base des mécanismes d'apprentissage de corrélations spatiales. Nous abordons le temps physique au paragraphe 5.2.2

Le second type de temps est ce que nous appellerons le *temps logique*⁶². Il s'agit du temps le long duquel s'articule la nécessité de *sérialiser* un comportement, comme nous l'avons évoqué au paragraphe 1.3.2. Il s'agit du temps qui sépare les causes des conséquences. Ce temps est beaucoup plus sujet à distorsion que le temps physique. En effet, attendre le bon moment pour serrer la main sur une balle que l'on saisit au vol suppose d'exploiter une temporalité fine, mais stable. C'est elle qui relève du temps physique et des traces. En revanche, attendre d'avoir trouvé ses clés avant de démarrer la voiture relève d'une succession logique d'étapes, et nombre d'entre nous expérimentent avec nervosité les distorsions temporelles considérables auxquelles cet enchaînement d'étapes est sujet. Nous soutenons au paragraphe 5.2.2 que le mécanisme corollaire au temps logique est l'activité *bistable*, qui nécessite l'occurrence d'un événement pour devenir active, et l'occurrence d'un autre pour s'éteindre.

Enfin, nous avons évoqué au paragraphe 3.6 le temps de la *relaxation* d'un système dynamique vers un point fixe. Le mécanisme corollaire en est dans notre domaine la *compétition* et la *mise à jour asynchrone*.

5.2.1 À quoi bon reconnaître les séquences ?

Avant d'entrer dans le vif du sujet sur la question du temps dans nos modèles, nous souhaitons nous attarder sur la notion de reconnaissance de séquences, qui lui est souvent associée (chaînes de Markov pour le traitement de la parole, typiquement).

Le limitation des algorithmes de traitement de séquences tient, selon nous, à la confusion qu'ils entretiennent entre la notion de séquence et celle de flux. Une séquence, ça a un début et une fin, alors qu'un flux est un objet infini dans le temps. Les interactions d'un agent situé avec son monde sont un flux, comme la parole, et il est parfois nécessaire de délimiter des séquences au sein de ce flux pour leur appliquer des algorithmes de programmation dynamique.

Sans renier les apports de tels algorithmes, il nous paraît artificiel de devoir déterminer un temps t_0 au sein du flux des interactions d'un agent avec son environnement. C'est un des reproches que nous adressons également au codage par rang proposé par Simon Thorpe [Thorpe and Gautrais, 1998], ainsi qu'au mécanisme de compétition distribué proposé par Kohonen [Kohonen, 1997], et à bien d'autres modèles. Nous nous sommes efforcé, et continuerons à l'avenir, de ne pas tomber dans ce « *travers* » du t_0 , au prix souvent de s'interdire certains raccourcis informatiques qui seraient parfois commodes.

Comment faire alors pour traiter un « *flux d'intelligence* », comme l'est un signal de parole que l'on écoute ? Lacan souligne que le sens n'est pas quelque chose qui se constitue en bijection avec le décours des symboles (i.e. les signifiants) de ce flux, mais qu'il y a un glissement entre les signifiants et le sens. En certains points du discours, le flux de sens et celui des signifiants s'accrochent, et le sens de la phrase se fait à rebours. Par exemple, si l'on dit « *Sa robe noire marque son appartenance...* » et que l'on termine par « *au folklore breton* », « *au Tribunal* » ou « *au troupeau* », on saisit ce mouvement à rebours qui attribue la sémantique associée à la robe. Lacan utilise l'image du « *capiton*⁶³ » pour rendre compte de cette propriété du décours des symboles et du décours de la sémantique de ne se rencontrer et de ne s'auto-déterminer qu'en certains points.

62. Nous empruntons la formule à Jacques Lacan, qui désigne par ce terme un concept très proche du nôtre.

63. Les points de capiton n'unifient deux tissus qu'en quelques endroits, le glissement étant permis entre ces points.

« Le point de capiton métaphorise cette propriété du langage qui veut qu'un signifiant, dans une chaîne parlée ne prend sens qu'après coup dans la mesure où c'est le dernier signifiant posé qui, rétroactivement, délivre le sens » .

Joël Dor, [Dor, 1985, page 138]

Nous interprétons ce phénomène d'après les éléments que nous avons présentés dans ce document. Selon nous, les phonèmes contribuent à constituer un état interne, comme le font les entrées d'un système dynamique à mémoire temporelle (cf. paragraphe 3.5). La succession des états du système dynamique, induite par le flux de phonèmes, est également sujette à un processus permanent de restitution de cohérence (cf. paragraphe 3.6), qui se cristallise, à l'instant où les conditions sont rassemblées, en un état cohérent, rendant compte de l'historique récent et en contexte du flux.

Trouver les mécanismes qui président à la constitution de ces instants où, ponctuellement, la sémantique se constitue, semble à portée de nos modèles, à condition d'abord la question du temps, qui pour l'instant leur fait défaut. En tout cas, ce processus est bien plus général que le phénomène langagier, puisqu'il peut inclure l'émergence d'une décision au cours d'un comportement. Il est de plus, par définition, un processus situé puisque le couplage avec le monde est permanent. Il nous paraît donc que mener une investigation de cette nature, qui implique de rejeter la démarche qui consiste à isoler une séquence pour la reconnaître, est la voie sur laquelle doivent naturellement s'orienter nos travaux de recherche.

5.2.2 Le temps physique dans les systèmes neuro-mimétiques

Certains traitements neuronaux requièrent une prise en compte de la valeur des écarts temporels entre différents événements. Ainsi, si l'on souhaite détecter un mouvement en en appréciant la vitesse, il convient de vérifier que l'événement survient à deux endroits, séparés du laps de temps correspondant à cette vitesse. C'est ce qui se passe au niveau du traitement des transitions de contraste en mouvement sur la rétine, qui est modélisé par un filtrage de Gabor spatio-temporel [Adelson and Bergen, 1985].

Lorsqu'il s'agit d'apprendre des corrélations temporelles, la plupart des algorithmes consistent à « recycler » l'apprentissage associatif hebbien connu pour les corrélations spatiales, en appliquant tel quel non plus sur les signaux eux-mêmes mais sur les traces de ces signaux [Sutton and Barto, 1981]. Ces traces sont en général des fonctions décroissantes, comme dans [Dominey, 1995] par exemple mais aussi dans nos propres travaux [Frezza-Buet and Alexandre, 2002], mais l'on peut envisager la capture de corrélations temporelles plus fines en mesurant la corrélation instantanée de traces plus complexes, comme celles de la figure 5.1. Des modèles inspirés des colonnes corticales, précurseurs de nos travaux, ont également utilisé cette notion de trace pour le traitement de la parole [Durand and Alexandre, 1996]

Dans le domaine des réseaux impulsifs, les travaux de Hopfield se basent également sur des traces. Il s'agit dans ce cas d'une fréquence d'émission des spikes qui diminue après le survenue d'un événement. Les vitesses de diminution sont variables dans le modèle, et des neurones détecteurs de synchronisation repèrent les traces qui, à un moment donné, sont de même fréquence, détectant ainsi des conjonctions temporelles [Hopfield and Brody, 2000; Hopfield and Brody, 2001].

Mentionnons enfin qu'il existe d'autres procédés, plus rares, que les traces pour prendre en compte le temps physique dans les réseaux de neurones. Nous pensons en particulier aux modèles

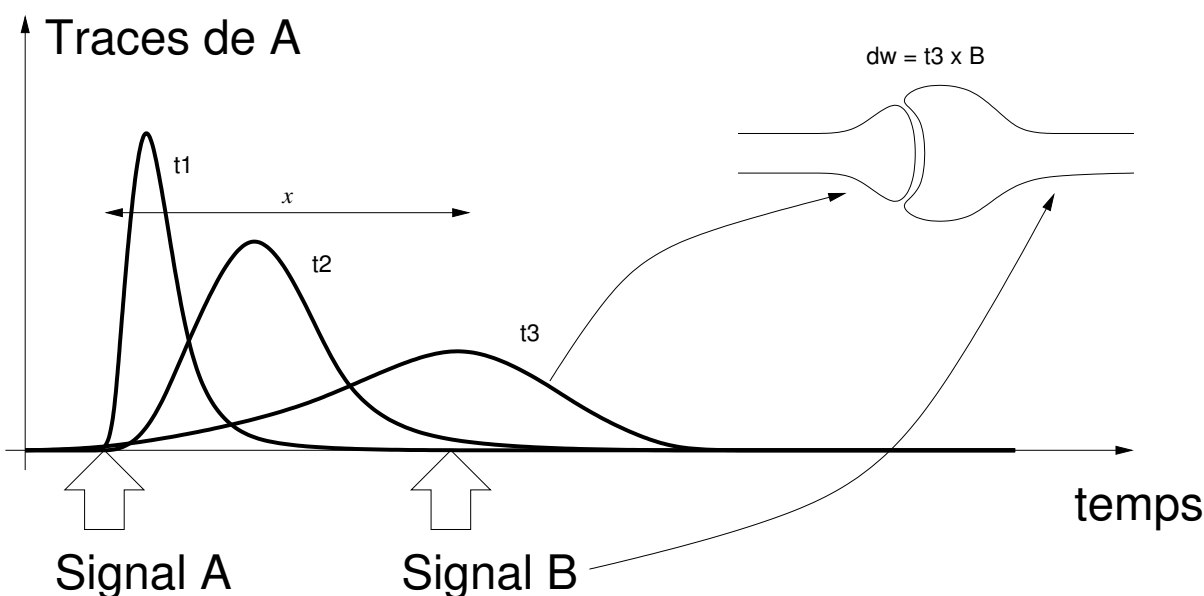


FIGURE 5.1 – La synapse se renforce en fonction de la corrélation entre le signal post-synaptique B et la trace, disons t_3 pour cet exemple, du signal A . Il en résulte que la synapse mesure la régularité de l'événement « A prédit B avec une antériorité de x secondes ». La forme des traces et leur rôle dans l'apprentissage sont inspirés de [Grossberg and Schmajuk, 1989].

inspirés du système auditif de la chouette, où c'est l'architecture du réseau, et non des traces, qui rend le système sensible aux déphasages des signaux captés par les oreilles gauche et droite [van Hemmen and Kempter, 1998].

Nous n'avons pas encore intégré dans nos modèles la prise en compte du temps physique, en fournissant sur les champs de neurones des activations qui soient la trace d'autres activations, voire en fournissant une batterie de traces de constantes de temps différentes, comme le suggère [Grossberg and Schmajuk, 1989] et comme on le trouve dans [Dominey, 1995]. La nature de l'auto-organisation temporelle résultante reste à découvrir, et certainement à comparer avec des modèles d'auto-organisation temporelle comme ceux de Wiemer [Wiemer, 2003].

Nous pensons toutefois que de tels mécanismes sont de première importance pour des centres tels que le cervelet ou la moelle épinière, directement en prise avec les automatismes. Nous n'avons cependant pas d'argument pour exclure ces mécanismes des traitements corticaux, ne serait-ce qu'à cause des traitements auditifs et des traitements du mouvement. De plus, indépendamment de la plausibilité biologique, l'intégration de ces mécanismes est une perspective qui, selon nous, est intéressante d'un point de vue informatique, et c'est ce que visent nos travaux.

La thèse démarrée en janvier 2010 de Bassem Khouzam, que nous co-dirigeons avec Frédéric Alexandre, aborde cette question. En effet, il s'agit de proposer un système dynamique récurrent (cf. paragraphe 3.5.3) pour capturer sous forme d'un état instantané l'historique récent des entrées reçues par le système. En cela, cet objectif se rapproche des travaux initiaux proposés dans [Dominey, 1995]. Toutefois, nous cherchons à faire en sorte que ce système récurrent s'auto-organise, en s'appuyant sur la logique du modèle `bi_jama`. Il s'agit alors de construire une

représentation qui, à chaque instant, caractérise de façon markovienne un flux d'entrée qui ne l'est pas forcément. C'est le rôle des réservoirs, puisque l'objectif est de pouvoir prendre à chaque instant, à partir de l'état du réservoir, une décision qui implicitement dépend de l'historique récent. La nouveauté de notre approche réside dans l'auto-organisation du réservoir, qui n'est plus un réseau aléatoire mais un réseau adaptatif.

5.2.3 Le *temps logique* dans les systèmes neuro-mimétiques

Durant nos travaux de thèse [Frezza-Buet, 1999], nous avons particulièrement mis en avant la notion d'*arbre d'appel* proposée par Yves Burnod [Burnod, 1989]. L'appel est, selon Burnod, un état d'activation intermédiaire de la colonne, qui n'est pas assez fort pour produire une action, mais qui peut quand même se propager au sein des colonnes. Le chemin de propagation des appels correspond à l'exploration parallèle, en chaînage arrière, des sous-buts nécessaires à l'obtention d'un but. Lorsque cet appel rencontre des colonnes excitées par l'état courant du cortex, la conjonction entre ce qu'il faut (appel) et ce que l'on a (excitation) produit des actions puis une transition d'états, jusqu'à « remonter » la chaîne de sous-buts et arriver au but initial.

Dans cette notion d'appel, il y a la notion d'étape, plus que celle de séquence. Ce qui est en jeu, c'est de savoir « à qui c'est le tour » de devenir d'actualité. Ce tour est attendu comme la prochaine étape, et non comme devant se produire dans x millisecondes, ce qui éloigne cette notion du temps physique.

Cette notion d'appel mérite toutefois d'être affinée, pour gérer la complexité de *sérialiser* le comportement [Guigon, 1993]. En effet, il est nécessaire de garder en mémoire, de façon *provisoire*, le contexte de l'activité courante (voir notre exemple de l'apiculteur page 21). Ce stockage en *mémoire provisoire*⁶⁴ [Fuster, 1997; Fuster, 2000] est supporté par des circuits récurrents, qui autorisent le maintien d'activités au cours du temps. Ces boucles ont alors des activités bistables : un événement les enclenche jusqu'à ce qu'un autre les remette au repos. L'ensemble des boucles actives correspond à l'état du système, qui peut de ce point de vue se comporter comme un système dynamique à mémoire temporelle (cf. paragraphe 3.5), avec la particularité que ce sont *des activités* et non des poids qui influent sur le comportement du système face à l'événement courant. Ce concept de stockage de contextes sous forme d'activités est au cœur de la modélisation de tâches imputées au cortex préfrontal [Dehaene and Changeux, 1989; Rougier and O'Reilly, 2002; Rougier et al., 2005].

La mise à disposition, pour le calcul cortical, d'un jeu de bistables est pour nous la condition de la gestion du temps logique par nos modèles, permettant une hiérarchisation de la notion d'appel en buts et sous-buts. Nous souhaitons revenir sur ces questions, que nous avons abordées durant nos travaux de thèse, fort de la formalisation opérationnelle du calcul cortical présentée dans ce document.

5.3 Récompense

Nous avons développé au paragraphe 5.2.3 la question du raisonnement logique articulé, pour lequel nous soutenons la pertinence de mécanismes à base de bistables, mais nous avons passé sous silence ce que nous pensons être *la* composante principale des raisonnements articulés, et de façon plus générale des comportements, à savoir la *récompense*.

64. *Provisional memory* en anglais.

Le cortex frontal, que nous voyons, d'après [Fuster, 1997] comme le siège de la gestion du temps logique, est aussi le pôle qui intègre l'information au regard de la récompense qu'elle peut apporter. Que la récompense soit un élément central du comportement chez les animaux est avéré, mais la question qui nous préoccupe est en quoi cette notion peut servir notre discipline, l'informatique.

En intelligence artificielle tout d'abord, on peut identifier des buts, et remonter, en chaînage arrière, les sous-but. L'identification des buts et la désignation des sous-but peut être considérée comme une récompense, dont l'anticipation se propage. La question est de savoir si cette réduction est pertinente au regard de ce qu'est la récompense. Avançons un peu en considérant l'*apprentissage par renforcement* [Sutton and Barto, 1998]. Dans les problèmes de type processus décisionnels de Markov⁶⁵ (MDP), la récompense est une valeur scalaire reçue par l'agent, qui construit une politique de choix d'actions afin d'optimiser l'espérance de *cumul* de cette récompense. On dépasse ainsi, dans la formulation, le contexte du chaînage arrière. Si l'on considère l'algorithme *Value Iteration* toutefois, cela revient également à propager une valeur au travers des états, depuis l'état but, mais cet algorithme est un cas particulier, pour lequel la matrice de transition est connue.

Si l'on se base sur le formalisme de l'apprentissage par renforcement, la récompense est *un* critère numérique, qui sert de base à un processus d'optimisation dont le comportement est le résultat. L'information de récompense est externe à l'agent. Dans le cas du chaînage arrière au contraire, il s'agit de suivre la chaîne des buts et des sous-but, et le succès d'un sous-but que l'on cherchait à résoudre est un signal interne. Nous l'avons appelé « *satisfaction* » dans nos travaux de thèses [Frezza-Buet, 1999], mais cette idée était elle-même issue des travaux de Pascal Blanchet [Blanchet, 1992; Blanchet and Alexandre, 1995].

Il nous apparaît donc nécessaire de faire cohabiter deux mécanismes. L'un, analogue à ce qui est fait en apprentissage par renforcement, fait intervenir une récompense externe, qui fait de l'agent qui l'intègre dans son comportement un agent situé. L'autre mécanisme est une récompense interne, guidant elle aussi des apprentissages, mais liée à la construction d'une logique dans l'articulation du comportement. Dit de façon imagée, la première relève du « miam ! » alors que la seconde relève de l'« eureka ! », et nos travaux passés ont entretenu une confusion entre les deux, puisque nous récompensions directement les buts.

Il est probable qu'une investigation de ces questions à la lumière du fonctionnement du cerveau, ainsi que du caractère comportemental du raisonnement, devrait conduire à unifier l'« *eureka!* » et le « *miam!* », mais cela passera certainement par une reconsidération de l'apprentissage par renforcement. En effet, ce dernier, dans les formulations actuelles, est surtout sensible au temps physique. Il est d'ailleurs illustré sur le problème de la construction de contrôleurs, comme le pendule inversé. Ces problèmes nous paraissent être bien plus du ressort de la logique du cervelet que de celle du cortex, plus inférentielle.

5.4 Ajout de mécanismes de récompense dans *bijama*

Nous avons présenté comme une extension à nos travaux la question du temps et de la gestion de la récompense, mais il s'avère que nous avons déjà commencé à étudier ce point, qui a fait

⁶⁵. *Markovian Decision Process* en anglais.

l'objet de la dernière année de thèse d'Olivier Ménard [Ménard, 2006]. Il s'agit de premiers pas, au regard des ambitions exprimées dans ce chapitre, mais ces études préliminaires nous permettent de présenter de manière plus instanciée la façon nous concevons ces extensions.

5.4.1 Le problème du *balbutiement moteur* sur un cas simple

On mentionne parfois le balbutiement moteur⁶⁶ comme une des voies d'acquisition d'une compétence motrice. Sans développer cette question sous l'angle de la psychologie expérimentale, nous dirons simplement que le balbutiement moteur consiste à produire des actions aléatoirement, à observer les contingences sensorimotrices qui en résultent, puis à *exploiter* l'information ainsi acquise pour réaliser des asservissements.

Nous avons tenté d'aborder cette question durant la thèse d'Olivier Ménard, du fait de notre implication dans le projet Avim (cf. paragraphe 8.1.4). Le problème que nous abordions est celui illustré sur la figure 5.2, qui consiste à jouer sur les angles des deux articulations d'un bras robot pour asservir la position de l'extrémité du bras. Plus précisément, notons $\theta = (\theta_1, \theta_2)$ la position du bras, que nous qualifierons de *proprioception* de notre système. Notons également $X = (x, y)$ la position, dans un repère cartésien, de la « main » dans l'espace de travail. On s'intéresse à apprendre, pour une position donnée du bras, les effets sur la position de la main d'une variation de position articulaire. Ceci s'exprime par l'équation 5.1 :

$$dX = \mathcal{F}_\theta (d\theta) \quad (5.1)$$

Comme discuté dans [Ménard, 2006] et défendu dans [Baraduc, 1999], apprendre \mathcal{F} plutôt que la relation directe entre X et θ comme cela est fait dans [Ritter *et al.*, 1992, chapitre 11], permet de pouvoir utiliser un outil sans nécessiter une remise en cause de ce qui est appris sans l'outil. La partie droite de la figure 5.2 illustre cette propriété, en montrant que ce qui change avec l'outil est simplement la proprioception.

Le problème posé par le balbutiement moteur est alors celui illustré sur la figure 5.3. Si, pour une proprioception $\theta = (\theta_1, \theta_2)$ donnée, on choisit $(d\theta_1, d\theta_2)$ uniformément dans une sphère centrée sur l'origine, on explore des positions de main en favorisant une direction (cf. partie gauche de la figure 5.3). Or pour apprendre un asservissement, il faudrait avoir la distribution des *commandes* utiles au comportement. Si on admet qu'il est pertinent de savoir déplacer la main dans toutes les directions, il faudrait, pour entraîner les algorithmes, fournir des variations $(d\theta_1, d\theta_2)$ tirées non uniformément, comme l'illustre la partie droite de la figure 5.3. Or pour générer cette distribution-là, il est nécessaire d'inverser la jacobienne de \mathcal{F}_θ , ce qui revient à connaître le modèle inverse du robot.

Les travaux de nos partenaires dans le projet Avim [Baraduc, 1999] ont nourri l'apprentissage avec des données issues du modèle inverse, ce que nous avons choisi de nous interdire⁶⁷.

*Le problème du bras robot a été très instructif pour nous. Toutefois, il s'agit d'une transformation moto-motrice selon nous, puisque la partie « visuelle » du problème est en fait un vecteur (dx, dy) et non l'activation d'une batterie de filtres, avec une organisation rétinotopique (cf. figure 4.4). Une extension de **bijama** à une prise en compte de la vision par des champs récepteurs organisés de façon rétinotopique fait l'objet de la thèse, en cours, de Lucian Alecu, que nous co-encadrons avec Frédéric Alexandre.*

66. *Motor babbling* en anglais.

67. Ce qui n'empêche pas de souligner ici que la notion de distribase et de codage monotone de la proprioception avancée par ces travaux est intéressante.

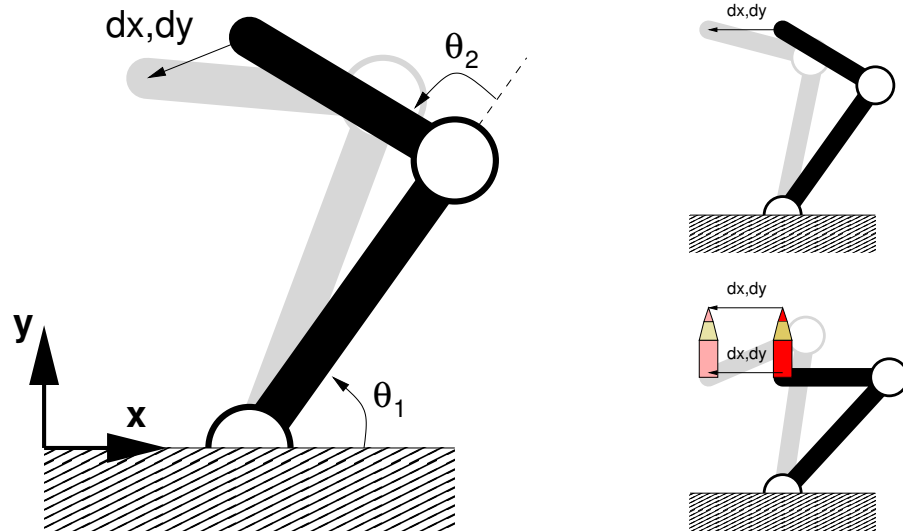


FIGURE 5.2 – À gauche, un bras robot à deux degrés de liberté (les articulations sont figurées par des cercles), dont on commande les angles θ_1 et θ_2 . On s'intéresse au déplacement (dx, dy) exprimé dans un repère cartésien, consécutif à une variation d'angle $(d\theta_1, d\theta_2)$. À droite, on voit que dans le cas de l'utilisation d'un outil pour réaliser le même déplacement, c'est la même fonction de contrôle \mathcal{F} qui est utilisée, mais avec une proprioception (θ_1, θ_2) « de départ » qui est différente dans les deux cas.

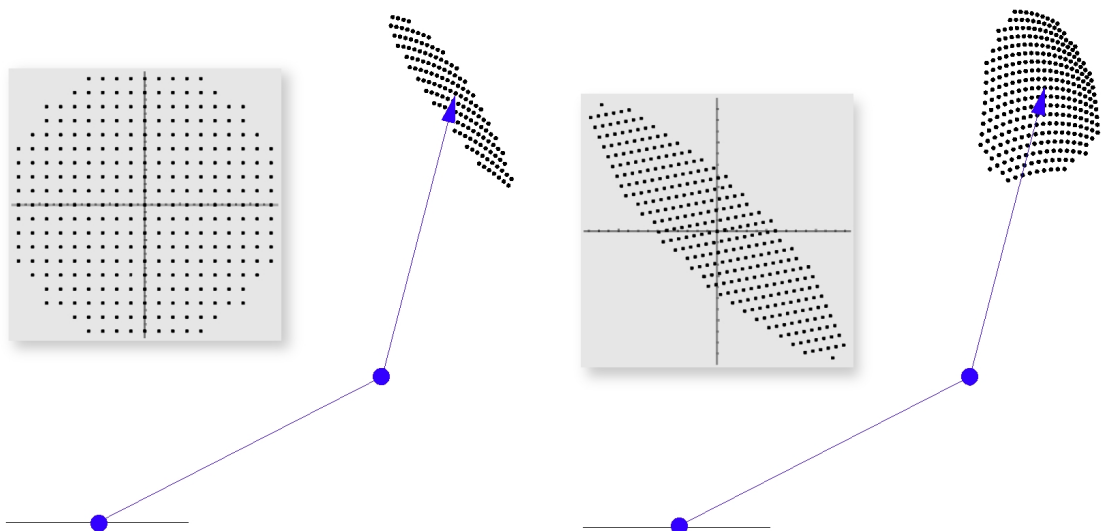


FIGURE 5.3 – Sur les figures est représenté un bras à deux articulations (disques bleus), dans une position donnée. Les encadrés illustrent des variations des deux angles, et on représente par des points en bout de bras l'endroit atteint quand chacune des variations est appliquée. Sur la figure de gauche, on voit que des variations uniformément distribuées, comme le seraient celles d'un balbutiement moteur, conduisent à explorer des positions de bras biaisées. À droite, on montre quelle devraient être les variations explorées par un balbutiement moteur pour avoir des positions de l'extrémité uniformément réparties, comme celles utilisées pour la commande. L'encart de la figure de droite est obtenu à partir du modèle inverse du bras.

L'objet du paragraphe suivant est de montrer comment nous avons abordé l'asservissement de ce bras avec le modèle `bijama`. Ces travaux sont toutefois des premières tentatives, et n'ont pas la qualité d'asservissement de travaux exploitant les cartes auto-organisatrices de Kohonen [Barreto *et al.*, 2003; Barreto and Souza, 2006]. Ces derniers travaux sont toutefois comparables aux nôtres du fait qu'ils font la promotion d'un *codage tabulaire*⁶⁸ dans l'asservissement, ce qui revient à tabuler la correspondance entre les coordonnées articulaires de la commande et les coordonnées cartésiennes de la position de la « *main* » dans l'espace de travail.

5.4.2 Architecture

L'apprentissage de l'asservissement par une architecture basée sur le calcul cortical tel que le définit `bijama` ne peut se contenter « *d'observer* » les régularités sensorimotrices générées par des mouvements aléatoires, comme nous l'avons montré au chapitre précédent. Si cela avait été possible, c'est-à-dire si la jacobienne de \mathcal{F}_θ était toujours, quel que soit θ proche d'une rotation, sans déformation, on aurait pu réaliser l'architecture de la figure 5.4. On aurait alors soumis à cette architecture des triplets $(\theta, d\theta, dX)$ obtenus en choisissant aléatoirement θ et $d\theta$, en observant le dX obtenu. Une fois les cartes organisées, il aurait suffi de présenter θ , la proprioception, dX , la consigne, et de laisser le système se relaxer, du fait des poids appris sur les prototypes des cartes (étage thalamique) et sur les *stripes* (étage cortical). L'état d'équilibre construirait une bulle sur toutes les cartes, celle sur la carte gérant les $d\theta$ correspondant alors, du fait de la *restitution de cohérence*, à l'action à réaliser pour satisfaire la consigne. Le problème soulevé par le paragraphe précédent est que les consignes présentées n'ont pas nécessairement été observées lors de l'apprentissage, pour lequel les $d\theta$ sont générés aléatoirement (cf. figure 5.3).

Face à cette difficulté, nous avons été amenés, avec Olivier Ménard, à introduire un mécanisme de récompense dans nos architectures [Ménard and Frezza-Buet, 2005; Ménard, 2006]. Nous l'avons fait comme illustré sur la figure 5.5, dont la légende présente le rôle des différents constituants de l'architecture.

La dynamique du système est alors la suivante. Lorsqu'on impose la proprioception $d\theta$ et la consigne dX , la mise en cohérence des modules, i.e. le processus de relaxation multi-cartes, produit une bulle sur la carte motrice. Cette bulle produit à son tour une activité d'entrée sur la carte spinale, *sur laquelle s'opère une compétition* afin de sélectionner une région de la carte spinale. Cette région détermine l'action effectuée par le système. Parallèlement à ce mécanisme, l'unité basale reçoit l'état des quatre modules corticaux et l'utilise pour prédire une récompense attribuée à l'action proposée. Si cette récompense prédite est forte, le système « *sait qu'il sait* », et l'action effectuée est celle sélectionnée dans la carte motrice. Si en revanche la récompense prédite est faible, le système « *sait qu'il ne sait pas* ». Dans ce cas, cette unité bruite l'entrée de la carte spinale, de sorte que l'influence de la carte motrice n'ait pas la capacité à forcer une action. La bulle se forme n'importe où, après un processus de compétition couplé avec celui de la carte motrice, du fait des connexions adaptatives réciproques entre les deux modules. La perturbation sur la carte motrice *se propage* aux autres cartes, puisque tous les processus de compétition sont interdépendants.

En cas de prédiction de récompense faible, la stabilité du système est perturbée de sorte à trouver un autre point d'équilibre, qui satisfait encore aux entrées dX et θ . Les degrés de liberté apportés par la carte associative et surtout par la carte motrice, qui n'est pas contrainte par l'extérieur, sont exploités lors de ce processus.

68. Voir page 84.

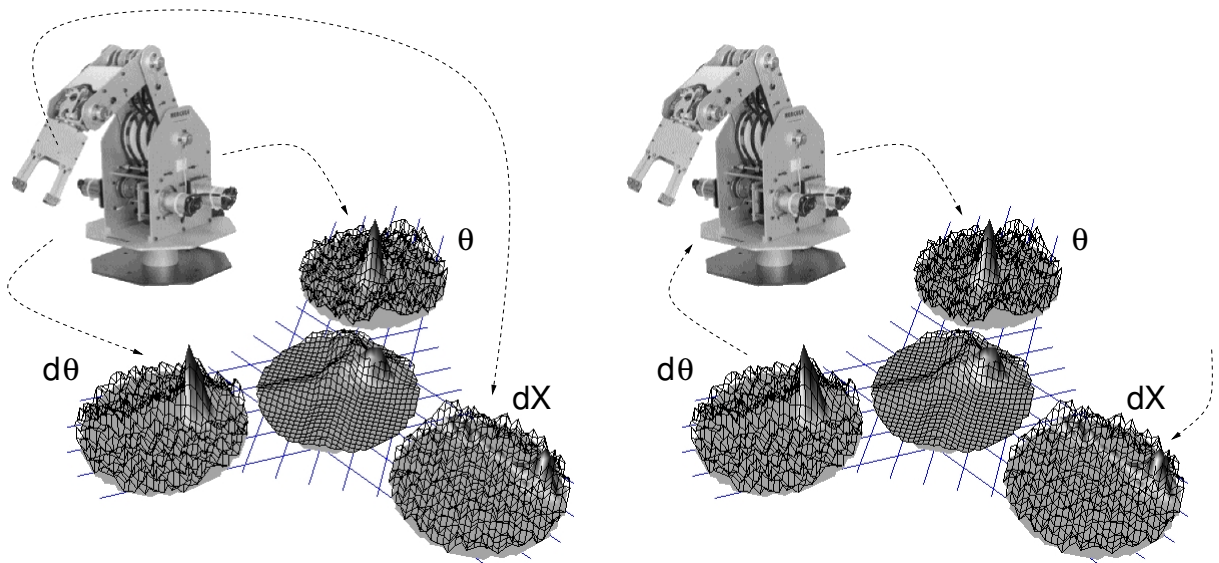


FIGURE 5.4 – L'apprentissage du modèle (figure de gauche) consiste à choisir une position θ du robot, ainsi qu'une variation $d\theta$, et d'observer dX qui est le déplacement de la main qui *en résulte*. Le triplet $(\theta, d\theta, dX)$ est alors présenté au système cortical. On réalise de nombreuses expériences ainsi construites. Une fois les correspondances entre ces modalités apprises on peut utiliser la capacité du système à rétablir la cohérence ainsi acquise en donnant au réseau la position θ , ainsi qu'une consigne de déplacement dX (figure de droite). Si l'on n'impose rien sur le module $d\theta$, c'est sur la variation d'angle cohérente avec θ et dX que se formera la bulle d'activité, définissant ainsi la commande. L'impossibilité de mettre en œuvre cette approche est discutée dans le texte.

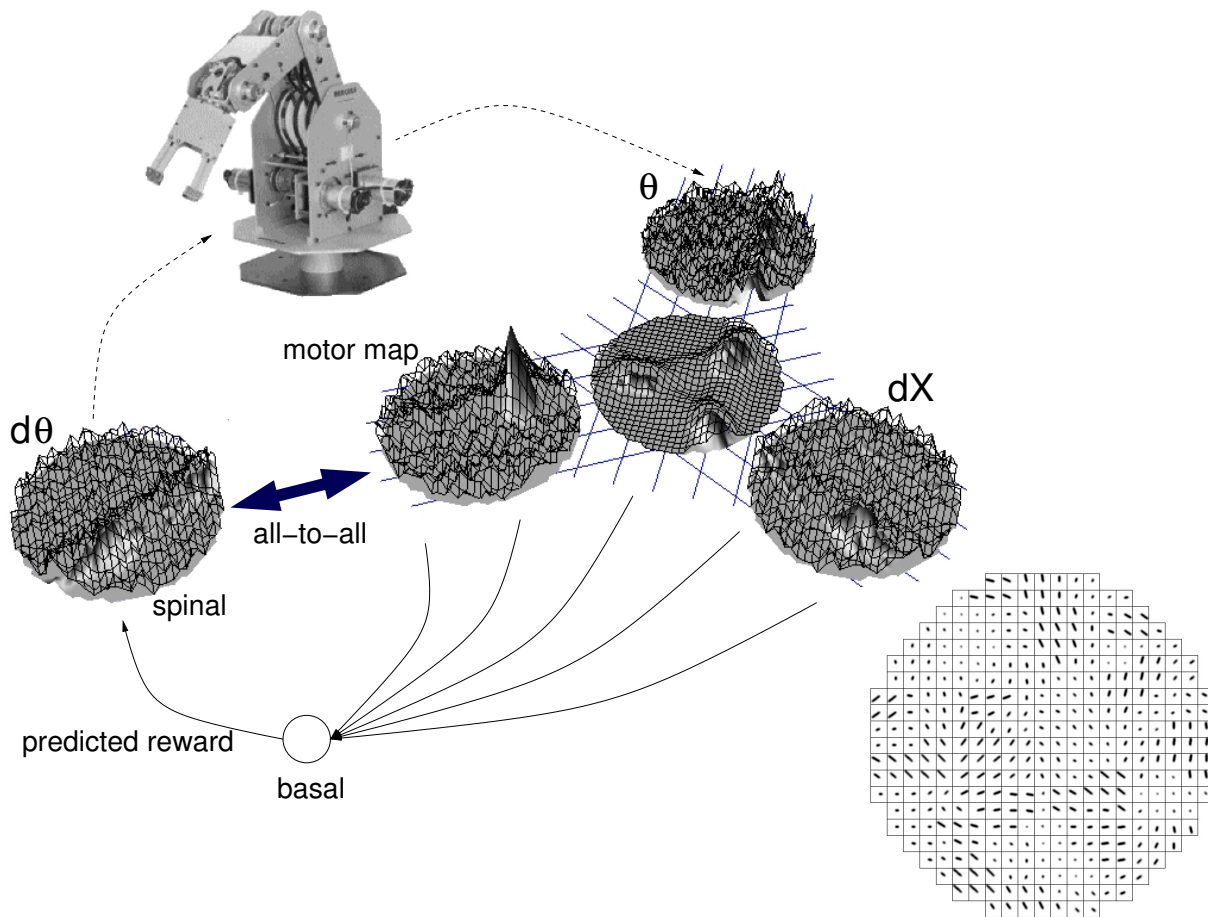


FIGURE 5.5 – Le système reçoit la position courante du bras θ , ainsi que la consigne dX , sous forme d'entrées de l'étage thalamique de deux des modules. Un module associatif met en relation ces deux modules avec un troisième module, moteur, dépourvu d'entrée thalamique. L'activité du module moteur est projetée sur un module « *spinal* », par une projection issue d'un apprentissage. Une activation dans le module spinal est directement câblée à une variation $d\theta$ de la configuration articulaire. Le système prédit la récompense de l'action courante à partir de l'état des modules corticaux, ce qui est le rôle de l'unité « *basal* ». Plus la récompense prédite est faible, plus cette unité bruite le module « *spinal* », de sorte qu'il choisisse une action aléatoire.

La différence avec le protocole illustré sur la figure 5.4 est fondamentale, et se résume comme suit.

Le système n'est plus témoin des contingences sensorimotrices, mais doit stabiliser la prise en charge des perceptions pour calculer une commande récompensée. On retrouve ce que nous avons mentionné page 15, à savoir que le système accède à la causalité de la récompense, ici réaliser la consigne, en agissant sur le monde, et non en capturant, passivement, les corrélations.

Si on laisse le système évoluer, sa capacité à avoir une récompense (elle est donnée en fonction du degré de réalisation de la consigne) augmente, et les cartes s'organisent pour cet objectif. Par exemple, la répartition des consignes sur la carte `dX` est illustrée sur la figure 5.5 (en bas, à droite). Du fait que les consignes sont imposées, et non plus calculées par le modèle direct du robot, elles sont par construction bien distribuées, et le problème évoqué sur la figure 5.3 ne se pose plus.

Un des points délicats de cette approche est la souplesse du champ de neurones qui gère la compétition de la carte spinale, ainsi que la forme de la prédiction de récompense réalisée par le module basal, pour laquelle une régression linéaire est un peu trop restrictive. Ces points ont été étudiés par Thomas Legrand lors de son stage de Master dans notre équipe [Legrand, 2006].

Acteur-critique

L'architecture précédemment évoquée exhibe une composante « *acteur* » et une composante « *critique* ». La composante acteur est l'ensemble des cartes corticales, qui aboutit à la proposition d'une action motrice. La qualité de cette action est évaluée au niveau de l'unité que nous avons nommée « *basal* ». Dans notre modèle, les deux processus acteurs et critiques sont distincts, et l'ensemble est stable quand l'acteur propose les bonnes actions et que le critique sait que ce sont les bonnes actions.

La séparation de l'acteur et du critique n'est pas présente dans tous les algorithmes d'apprentissage par renforcement, puisque les plus classiques d'entre eux sont souvent des *critiques purs*, à l'instar de Q-Learning et Sarsa. Or les critiques purs calculent une fonction de valeur, qui approche celle du comportement optimal, mais la politique n'est que déduite mécaniquement de cette fonction de valeur⁶⁹. Pour certains problèmes, il arrive que l'on approche correctement la fonction de valeur, mais que la politique déduite soit très différente de la politique optimale. C'est pourquoi disposer d'un acteur indépendant du critique (même si il en tient compte) est un gage de qualité du comportement obtenu. En effet, si ce comportement est bien évalué par le critique, c'est qu'il est effectivement bon.

Nous avons, dans le modèle que nous proposons, extrait le rôle de critique de l'activité corticale (unité « *basal* »), mais le mécanisme de critique est encore frustré. Cela conduit à s'intéresser aux structures sous-corticales impliquées dans ces fonctions chez le primate, à savoir les ganglions de la base. L'extension de nos travaux dans cette direction est présentée au paragraphe 10.3.2.

5.4.3 Conclusion

La version présentée ici du caractère situé du calcul cortical tel que l'instancie `bijama` est par bien des aspects très simpliste. Nous pensons toutefois qu'elle jette les bases de principes

69. La politique est celle qui est gloutonne sur les valeurs calculées par les algorithmes.

nouveaux et de nature à guider nos recherches futures. Nous discutons donc d'abord, dans ce paragraphe, de ce qui est simpliste dans ce modèle, pour montrer ensuite la portée de cette approche, en dépit des raccourcis que nous faisons, provisoirement, aujourd'hui.

Le premier point qu'il convient de discuter est l'adéquation de notre architecture avec le système sensorimoteur des primates. Il est clair que notre proposition n'a pas vocation à répondre à cette question. Nous cherchons plutôt à aborder la question de la récompense et de sa capacité à nous affranchir de nourrir les apprentissages à l'aide d'exemples générés via le modèle inverse du système.

Le deuxième point est l'extrême simplification des mécanismes de gestion de la motricité (le module spinal) et de la prédiction de récompense (l'unique unité basale). En ce qui concerne l'unité basale, que nous nommons ainsi en référence aux ganglions de la base, une étude plus approfondie permettra de décomposer sa fonction de sélection de l'action [Mink, 1996] et de prédiction de récompense [Schultz, 2001], puisque dans les ganglions de la base, de multiples centres nerveux se coordonnent à cet effet. Nous pourrions alors prétendre à instancier, par nos modèles, des conceptions plus intégrées des différents rôles des grands centres nerveux du cerveau, comme celles proposées par Doya [Doya, 1999].

Le troisième point est le caractère encore trop instantané de la gestion de la récompense, et des délais qui peuvent intervenir dans son occurrence. Aujourd'hui clairement, nos modèles n'intègrent pas les finesses d'un apprentissage par renforcement tel qu'il est présenté dans [Sutton and Barto, 1998].

Quatrièmement, nous avons exprimé le comportement du module basal comme un prédicteur de récompense, permettant d'exprimer que le système « *sait qu'il sait* » ou « *sait qu'il ne sait pas* » produire la récompense par ses actions. Il y a un tiers état qui n'est pas géré dans le modèle, qui est celui de la confiance en la prédiction, et qui mériterait d'être approfondi au niveau des relations entre l'unité basale et le module spinal.

Malgré ces limitations, cette approche nous paraît être un premier pas qui répond aux points suivants, souvent épineux dans les modèles informatiques. Tout d'abord, l'introduction de la récompense n'est pas diffuse dans toute l'architecture, et n'a pas occasionné une refonte du modèle de calcul présenté au chapitre 4. L'effet global de l'apprentissage par renforcement tient, dans cette approche, aux perturbations d'équilibre qui se propagent dans le réseau, du fait de la *restitution de cohérence* multimodale qui fait la force du modèle *bijama*. Nous affirmons également que, dans cette architecture, les modules moteurs sont identiques aux autres modules, si ce n'est qu'ils n'ont pas d'étage thalamique, comme n'importe quel module qui n'est pas un module primaire. Cette uniformité est un de nos objectifs, puisque nous visons des modules génériques. Les modèles qui nous ont conduit à *bijama* étaient, eux, tributaires du besoin de faire une exception à la structure générale pour leurs constituants moteurs [Blanchet, 1992; Frezza-Buet, 1999].

*Même si aujourd'hui l'extension de **bijama** aux traitements situés n'est qu'un premier pas, nous pensons que cette extension est ouverte et trace la voie des développements futurs de nos recherches. À moyen terme, Lucian Alecu, lors de sa thèse ^a, tente d'aborder une structuration de la vision par l'exigence comportementale de l'agent, sur la base des travaux présentés ici, donnant d'emblée aux traitements visuels « primaires » une sémantique, ce qui est conforme avec les ouvertures de nos travaux vers le raisonnement situé prédicatif, comme nous l'avons discuté au paragraphe 5.1.*

a. Nous co-encadrons cette thèse, dirigée par Frédéric Alexandre

Deuxième partie
Environnement

Chapitre 6

Activités de recherche

« collatérales »

Il s'est présenté durant notre activité de recherches des opportunités autres que celles qui s'inscrivaient directement dans les objectifs que nous nous étions fixés. Les discussions de pause-café, par exemple, avec des collègues d'autres disciplines, ne sont souvent guidées que par une simple curiosité sur des domaines que nous ne connaissons pas. De même, la nécessité de réaliser une démonstration pour des activités de communication, comme par exemple la fête de la science, amène à mettre en place des réalisations ponctuelles en marge de notre projet de recherche.

La plupart de ces papillonnages sont éphémères, ce qui n'enlève rien au plaisir que l'on peut y prendre, mais parfois la mayonnaise prend, et nous nous sommes retrouvés à deux reprise dans la situation où il nous est apparu pertinent de faire un effort supplémentaire pour concrétiser ces discussions en une réalisation scientifique de plus grande ampleur. Il en coûte certes un léger détournement du temps de recherche vers ces projets collatéraux, mais nous pensons qu'il convient de laisser un peu de « *bruit* » dans une activité scientifique. L'expérience nous a donné raison, dans le cadre du projet ESCAPaDE, né de ce bruit, puisqu'il a permis de monter le projet InterCell qui lui, est clairement au service de notre projet de recherche fondamental.

Nous présentons donc dans ce chapitre les deux activités de recherche collatérales qui ont vu le jour en marge de notre projet. La première est la construction d'un *solver* numérique d'équations aux dérivées partielles, à savoir le projet ESCAPaDE dont nous venons de parler. La seconde activité est la conception d'un algorithme de quantification vectorielle, GNG-T, qui permet de gérer les distributions non stationnaires.

Ces deux projets font l'objet de publications. Pour ESCAPaDE, la soumission d'un article de journal est en cours, le papier est disponible sur HAL [Fressengeas and Frezza-Buet, 2006]. Pour GNG-T, nous avons publié nos travaux [Frezza-Buet, 2008], et d'autres collègues de notre équipe les utilisent dans le cadre d'applications médicales.

A posteriori, il convient de constater que les raisons pour lesquelles la « *mayonnaise a pris* » pour ces deux activités-là ne sont pas le fruit du hasard. ESCAPaDE est l'application à la physique du calcul cellulaire à grain fin, que nous défendons dans notre projet de recherche, et GNG-T est un algorithme de quantification vectoriel, ce qui est cohérent avec la promotion du *codage tabulaire* que nous faisons dans notre recherche (cf. page 84).

6.1 Résolution d'équations aux dérivées partielles

Nous avons, avec Nicolas Fressengeas, Professeur de physique au LMPOS, conçu un système de résolution d'équations aux dérivées partielles par automates cellulaires à grain fin. Nous avons déjà évoqué au paragraphe 3.3.3 les *Cellular Neural Networks* (CNN), qui ont été utilisés pour résoudre des équations aux dérivées partielles (EDP). La méthode est alors la suivante. L'EDP correspondant à un problème physique, elle contient des dérivées en espace et en temps. Le principe est de discrétiser l'équation suivant l'espace, à l'aide de différences finies, afin d'obtenir un système d'équations différentielles ordinaires, en temps uniquement. Ce système définit un CNN, dont l'évolution temporelle est censée suivre l'évolution temporelle du phénomène physique, ce qui n'est pas garanti.

L'approche que nous proposons avec Nicolas Fressengeas est un peu différente. Elle consiste à discrétiser par différences finies le problème suivant toutes ses dimensions, espace et temps. Les variables d'espace et de temps sont alors traitées de façon équivalente par cette approche.

Une fois le problème discrétisé, on dispose d'un maillage (x, y, z, t) dont les points sont des cellules. L'application des différences finies sur le système d'EDP à résoudre conduit à exprimer, en chaque point du maillage, un jeu de formules, une formule par équation du système. Ces formules, en un point, ne font intervenir que les valeurs des points d'un voisinage induit par les différences finies. Si l'équation est vérifiée par les valeurs en chaque point, ces formules doivent produire une valeur numérique nulle quel que soit le point du maillage considéré.

Ce jeu de formules décrit un CNN qui calcule, pour un état donné des valeurs du problème, l'erreur commise par chaque équation différentielle discrétisée en chaque point du maillage. Cet automate n'a d'autre intérêt que celui de vérifier a posteriori qu'un jeu de valeurs est ou non solution du problème.

Ce que nous avons proposé par la méthode ESCAPaDE, c'est de considérer, pour une répartition quelconque des valeurs, l'erreur qui correspond à cette répartition particulière. En chaque point du maillage, on peut calculer l'erreur commise en prenant la somme des carrés des valeurs des formules, ces valeurs étant censées être toutes nulles si l'équation est vérifiée en ce point. L'erreur globale est alors simplement la somme des erreurs en chaque point.

Nous avons montré, et nous renvoyons le lecteur à [Fressengeas and Frezza-Buet, 2006] pour une description formalisée de ce que nous présentons ici, qu'il est possible d'appliquer une méthode de Newton pour descendre le gradient d'erreur global. La descente que nous proposons est stochastique. Nous choisissons un point du maillage, et modifions ses valeurs de sorte à descendre le gradient. Le problème est qu'il ne suffit pas de réduire l'erreur locale, due au fait que les formules en ce point ne sont pas nulles, car modifier les valeurs d'un point a aussi des répercussions sur les valeurs des formules des points voisins, qui utilisent ces valeurs. La modification des valeurs d'un point a donc des conséquences sur un voisinage du point.

Nous avons montré que le gradient partiel de l'erreur globale, si l'on n'autorise que les variations des valeurs en un seul point, ne s'exprime qu'en fonction des valeurs du voisinage de ce point. Il en découle que l'on peut trouver la modification à appliquer aux valeurs d'un point en ne considérant que les points de son voisinage. Ainsi, la méthode de descente de gradient stochastique de l'erreur globale peut être implémentée sous forme d'un automate cellulaire évalué de façon asynchrone.

Pour un problème donné, avec ces conditions aux limites, la règle de mise à jour en un point peut être assez complexe, et nous la dérivons à l'aide d'un calculateur formel, en l'occurrence *Mathematica*. Ainsi, au sein de l'automate cellulaire, on détermine automatiquement un jeu de mises à jour, dont nous avons montré qu'elles restent locales. Tous les automates éloignés de conditions aux limites ont une mise à jour identique, et le système dérive autant de cas

particuliers que nécessaire pour les automates à proximité des conditions aux limites.

Pour résumer, il suffit de fournir à **Mathematica** l'expression discrétisée d'une EDP, par des méthodes de différences finies par exemple, et de déclarer quels sont les points du maillage qui sont des conditions aux limites. Ensuite, un package **Mathematica**, nommé **Dempf** et écrit par Nicolas Fressengeas, se charge d'appliquer une méthode de Newton sur l'erreur globale, afin d'en déduire les formules de mise à jour en chaque point du maillage. Une fois qu'un jeu de mises à jour a été identifié, et que l'on sait en quel point appliquer quelle mise à jour, **Dempf** génère des fichiers texte décrivant l'automate cellulaire qui est *dédié* au problème.

Le logiciel **simpf** que nous avons écrit (cf. paragraphe 9.6.5) traduit ces fichiers texte en code **C++** invoquant la bibliothèque **grumpf**, et l'on dispose ainsi d'un automate cellulaire *parallèle*, visualisable par des clients graphiques, qui est *ad hoc* au problème que l'on souhaite résoudre. Le caractère *ad hoc* n'est pas gênant puisque la procédure de production du logiciel est entièrement automatisée.

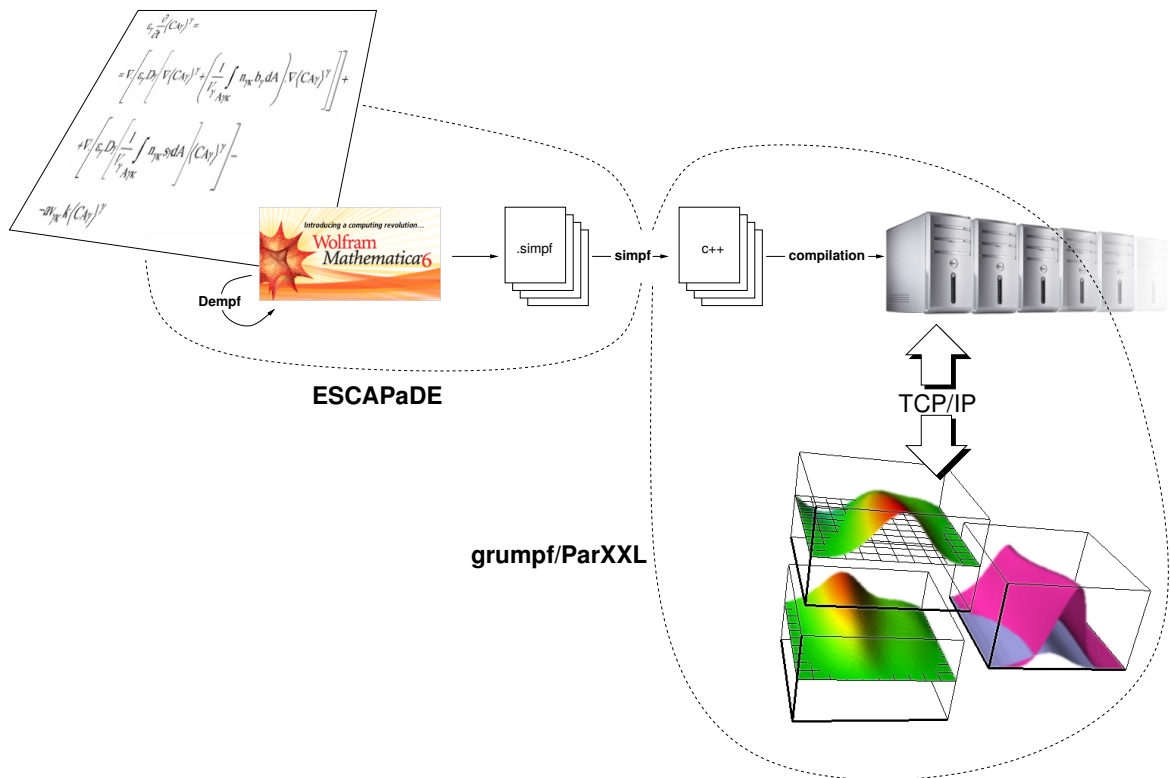


FIGURE 6.1 – De gauche à droite : le modélisateur saisit une équation aux dérivées partielles dans **Mathematica**, et le module **Dempf** génère une description d'un automate cellulaire dédié à sa résolution numérique, sous forme de fichiers **.simpf**. Le logiciel **simpf** convertit ces fichiers en du code **C++**, si bien que l'on obtient un serveur **grumpf**, capable de s'exécuter sur machines parallèles et interrogeable par des clients graphiques déportés.

L'ensemble de cette procédure est résumée par la figure 6.1. Notre approche permet aux physiciens de pouvoir réaliser des prototypes, lors de la mise au point de leurs modèles, qui soient immédiatement testables sur machines parallèles avec des outils de visualisation avancée. C'est la raison pour laquelle **ESCAPaDE** est le chaînon qui permet aux physiciens d'entrer dans la problématique du calcul à grain fin interactif défendue dans le projet **InterCell** que nous coordonnons (cf. paragraphe 8.1.2).

Depuis juin 2009, l'ensemble de cette suite logicielle a été porté sur le cluster `InterCell`. Les services de parallélisation et d'interactivité initialement assurés par `grumpf` sont fournis désormais par la bibliothèque `booz` (cf. paragraphe 9.7). Pour la partie formelle, Nicolas Fressengeas a abandonné `Mathematica` pour le logiciel libre `sage`. L'ensemble de la procédure, pour le cluster, est décrite sur le site du projet [InterCell, 2008].

6.2 GNG-T : Quantification vectorielle de données non stationnaires

Afin d'analyser une scène vidéo dans le cadre d'un contrat industriel, nous avons testé, avec des étudiants de Supélec de troisième année, l'opportunité d'utiliser l'algorithme *Growing Neural Gas* (GNG) de Fritzke [Fritzke, 1995] dans la procédure suivante. Le principe est résumé sur la figure 6.2. Tout d'abord, on effectue un filtrage pour différencier les objets présents du fond de la scène. Ensuite, on extrait les contours des objets, ce qui donne une collection de points. En appliquant GNG, on peut alors résumer la distribution des points des contours par un graphe. On réalise ainsi le lien entre une structure numérique, la distribution des points, et une structure symbolique, le graphe, dont on peut compter les cycles, les composantes connexes, etc.

Au cours de ce projet, nous avons constaté que GNG était très sensible au bruit, et supportait mal l'extension à des distributions non-stationnaires, comme le sont celles traitées sur notre flux vidéo. L'extension proposée par Fritzke pour la non-stationarité [Fritzke, 1997a; Fritzke, 1997b] repose sur un critère d'arrêt lié au nombre de prototypes, ce qui n'est pas satisfaisant non plus. De plus, de nombreux paramètres, tels que des traces décroissantes, n'ont pas de sens statistique précis.

Nous avons par conséquent cherché à approfondir la question de la quantification vectorielle d'une distribution non stationnaire, gardant le principe de construction d'un graphe qui approche la triangulation de Delaunay des prototypes formulée dans [Martinez and Schulten, 1994], que GNG exploite. Ce fut l'objet du stage de fin d'étude de Georges Adrian Drumea [Drumea, 2006], qui a conduit à une publication [Drumea and Frezza-Buet, 2007]. Suite à ces travaux, nous avons porté à maturité un algorithme, GNG-T, présenté et analysé dans [Frezza-Buet, 2008], dont nous décrivons ici quelques principes. Cet algorithme est disponible sous forme d'un *package* à base de patrons de classe C++, il s'agit de la bibliothèque `gngtlib` que nous fournissons sur notre site (cf. paragraphe 9.8.2).

Le principe est de considérer qu'à l'équilibre, lors d'un processus de quantification vectorielle, l'espérance d'erreur est la même sur toutes les cellules de Voronoï des prototypes, nous la noterons E . C'est pourquoi ces algorithmes sont sensibles à la densité de la distribution qu'on leur soumet. Ainsi, si la distribution est dense, la cellule de Voronoï accumule fréquemment de petites erreurs⁷⁰, alors que si la distribution est peu dense, la cellule de Voronoï accumule rarement de grandes erreurs. Comme E est constant pour les deux types de cellule, on comprend que la première soit beaucoup plus petite que la seconde.

Si maintenant on mesure la moyenne des écarts au prototype au sein de chaque cellule, indépendamment de la fréquence des prototypes (c'est ce que calculent les variables d'accumulation de GNG), on constate que les cellules des régions peu denses ont une forte variance, i.e. elles sont étalées, alors que les cellules où la distribution est dense ont de petites variances, i.e. les nombreux exemples sont serrés. Le principe de notre algorithme est d'échantillonner la valeur E du graphe courant, et d'ajouter ou retirer des prototypes pour que cette valeur s'asservisse à un

70. une erreur est l'écart entre un échantillon et le prototype

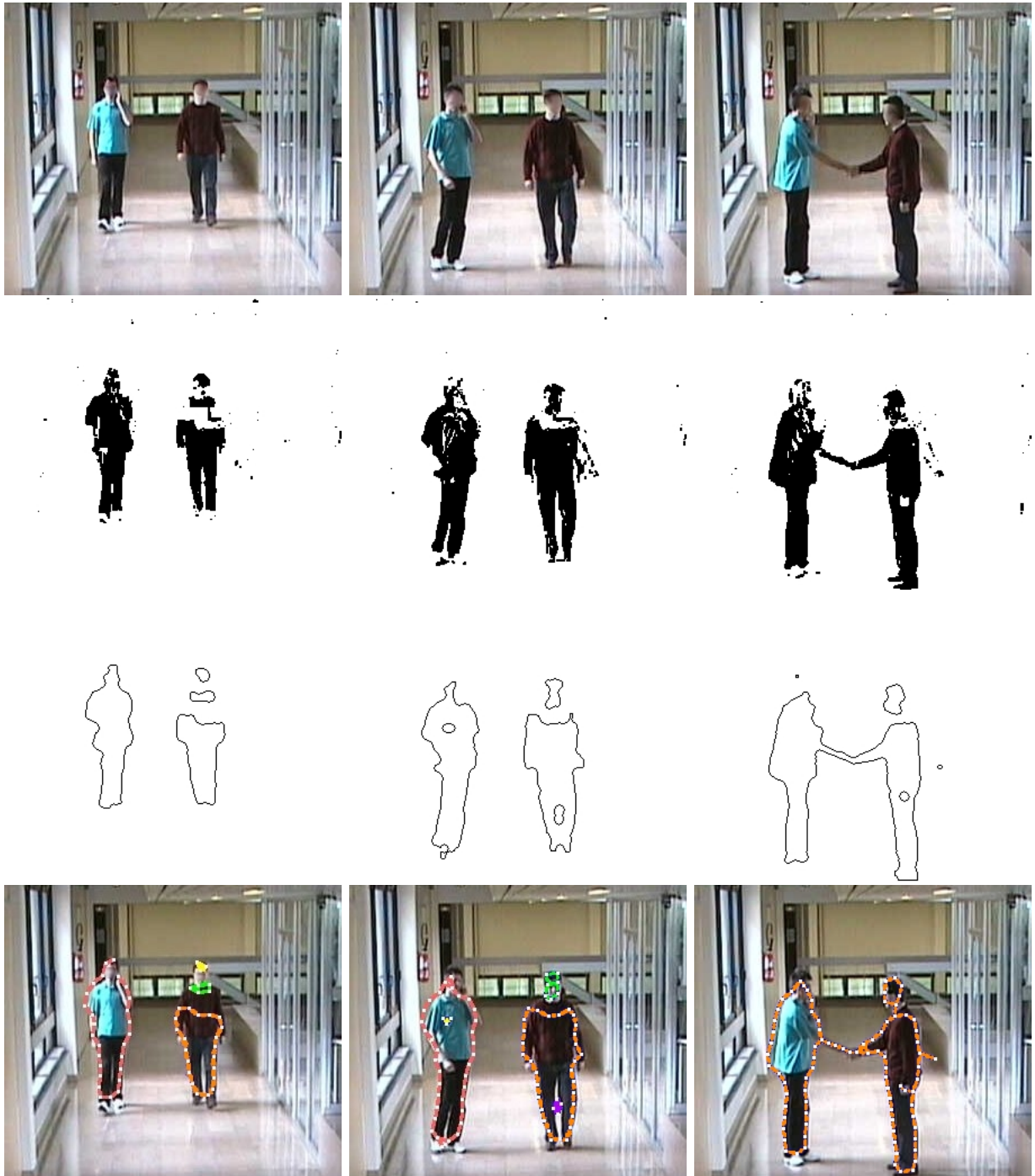


FIGURE 6.2 – La séquence vidéo est analysée comme suit. On extrait d’abord les pixels qui n’appartiennent pas au fond, et l’on nettoie l’image obtenue par des filtres booléens simples. Ensuite, on peut appliquer des opérateurs morpho-mathématiques, puis extraire les contours des régions obtenues. Le graphe qui résume ces points, obtenu ici par l’algorithme GNG-T, reflète la composition de la scène. D’après [Frezza-Buet, 2008].

paramètre T , fixé par l'utilisateur. Plus T est faible, plus la quantification vectorielle sera fine, i.e. plus le nombre de prototypes sera grand.

La figure 6.3 montre l'utilisation de l'algorithme sur une distribution changeante, la valeur T restant fixée. C'est donc au nombre de prototypes de s'adapter pour garder la même qualité de quantification.

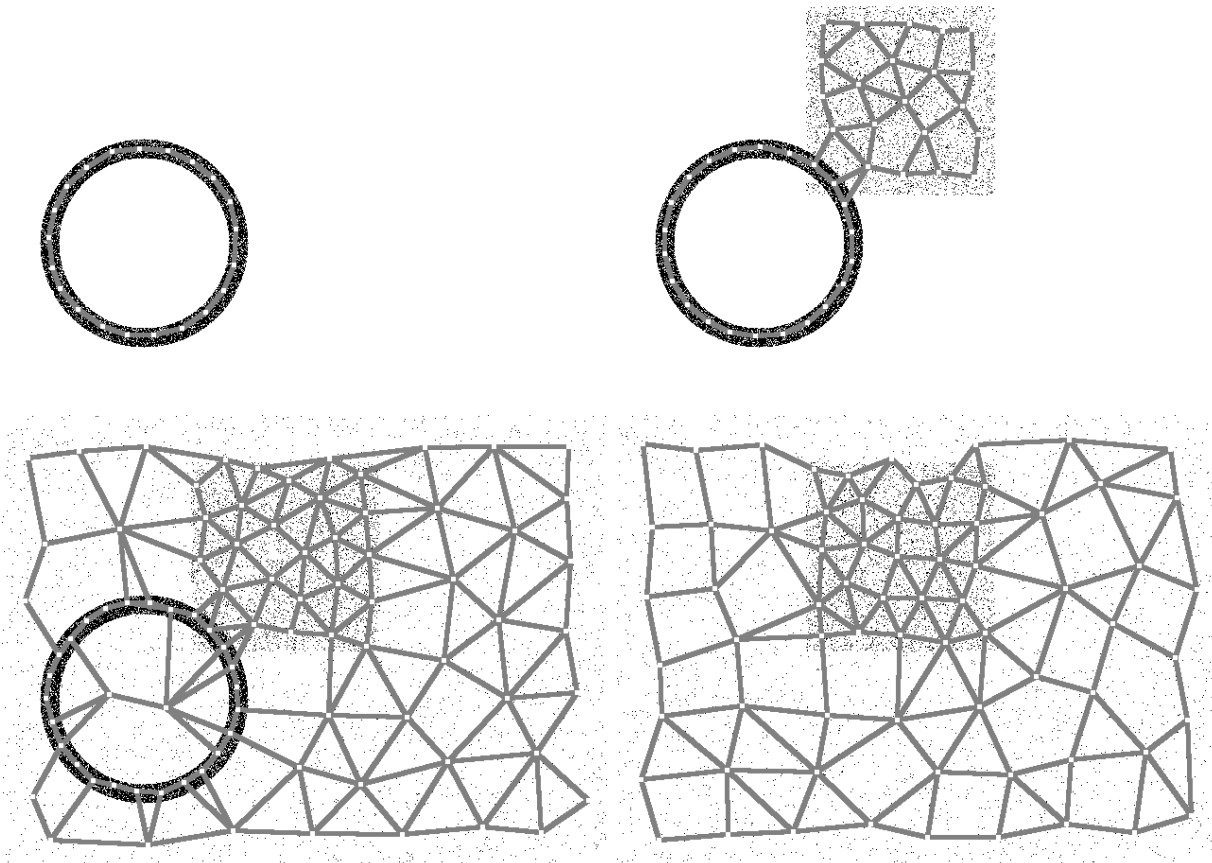


FIGURE 6.3 – Le graphe construit par GNG-T s'ajuste aux variations de la distribution, assurant une qualité de quantification vectorielle constante. Dans le sens de lecture sur les figures, on présente d'abord une couronne dense au réseau, puis on ajoute un carré un peu moins dense, puis du bruit. La dernière modification de la distribution consiste à retirer la couronne.

Il est plus intéressant de s'intéresser aux histogrammes de la variance des prototypes. On retrouve sur un histogramme de variance les différentes densités de probabilité (cf. figure 6.4). L'algorithme peut alors être autorisé à quantifier du bruit, il suffit de retirer a posteriori les prototypes de forte variance, ce qui a été fait sur la dernière ligne de la figure 6.2.

On peut constater, sur des vidéos, qu'en cas de modification progressive de la distribution, d'une image à l'autre, ce sont les mêmes prototypes qui sont associés aux régions au cours de leur mouvement. C'est l'effet de suivi, qu'assure notre algorithme. En revanche, si quelque chose apparaît, de nouveaux prototypes sont alloués, pour restaurer une quantification de qualité T .

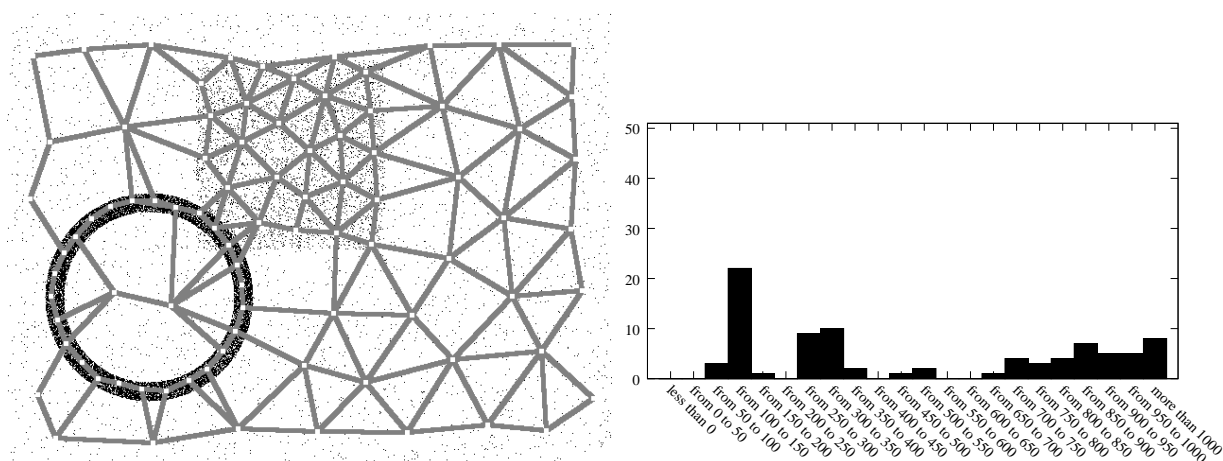


FIGURE 6.4 – Histogramme des variances. Le pic de gauche correspond aux prototypes qui sont placés sur le cercle, qui est dense. Le pic du milieu correspond aux prototypes du carré, et le pic plus étalé de droite correspond au bruit. On constate que l’histogramme permet de déterminer un seuil, ici autour de 600 pixel², au-delà duquel on peut considérer qu’un prototype représente du bruit.

La stabilité de l’algorithme est donc assurée par la mobilité des prototypes existants, qui suivent les changements continus de la distribution, mais la plasticité est également assurée en cas de changements brusques. Cette dernière est le fruit d’une stabilité d’un autre ordre de l’algorithme, qui en est la principale nouveauté, à savoir l’asservissement de la qualité de quantification.

L’algorithme est relativement simple, et robuste, comme le montre l’étude faite dans [Frezza-Buet, 2008]. Nous avons montré également, avec Olivier Pietquin et Jean-Louis Gutzwiller, que cet algorithme peut s’appliquer à une procédure en ligne d’identification non supervisée de locuteurs dans un signal de parole [Gutzwiller *et al.*, 2010].

Chapitre 7

Activités d’enseignement

Nous avons, depuis que nous avons entrepris, en 1996, de travailler dans la recherche, concilié une activité de recherche scientifique avec des activités d’enseignement supérieur.

On entend parfois parler de la mission d’enseignement des enseignants-chercheurs comme d’une charge, qui handicape l’activité de recherche. Il y a, à notre sens, quelques raisons à cela. La première est que l’on évalue plus quantitativement la recherche, depuis l’apparition d’indicateurs bibliométriques, et qu’effectivement, au regard de l’optimisation de ces indicateurs-là, enseigner c’est perdre du temps que l’on aurait pu investir dans la production scientifique. Une autre raison est que, pour les maîtres de conférences, 192 heures équivalent TD par an représentent un volume substantiel, auquel il faut ajouter les temps de préparation, de surveillance d’examen, de participation à des jurys, des réunions pédagogiques, etc. Ces contraintes fragmentent l’activité de recherche, effectivement.

Dans notre cas, à Supélec, nous avons eu une charge d’enseignement qui s’est alourdie progressivement, et qui, d’un point de vue purement comptable, occupe du temps que l’on aurait pu investir dans la recherche. Toutefois, nous souhaitons souligner dans ces quelques lignes les apports des activités d’enseignement pour notre projet de recherche. En effet, outre l’apport personnel lié à la situation d’enseignement, du fait des interactions avec les étudiants, la formation que nous dispensons, bien que technologique, a des retombées sur nos recherches plus fondamentales. Tout d’abord, et nous le mentionnons en introduction au chapitre 8, le temps d’enseignement, à Supélec, est l’occasion de s’ouvrir à des réalisations plus pratiques, lors de contrats industriels qui impliquent les étudiants. Cette expérience est un atout pour connaître le milieu industriel, et mesurer l’effort nécessaire, lors d’une valorisation de la recherche, pour passer d’un prototype à une solution exploitable par une entreprise. La démarche qui nous a conduit à concevoir l’algorithme GNG-T (cf. paragraphe 6.2) est issue d’une activité industrielle. De plus, nous sommes à Supélec en capacité d’affecter des élèves sur nos projets de recherche, auxquels ils apportent de la « *main d’œuvre* » précieuse. Il faut pour cela faire l’effort d’isoler dans nos besoins de développement des modules abordables par les étudiants, puis en assurer l’intégration effective après réalisation. Bien que cette démarche ne soit pas systématiquement couronnée de succès, nous avons, dans nos réalisations, pu profiter des capacités des élèves de Supélec (voir chapitre 9). Enfin, l’enseignement a cet avantage de nous « *forcer à approfondir* » des notions collatérales à nos domaines de recherches, cela a été le cas pour nous pour les méthodes à noyaux et la théorie de la généralisation, mais aussi dans la programmation C++ avancée (patrons de classes et méta-programmation) que nous enseignons et exploitons dans la mise en œuvre de nos modèles, de même que la visualisation 3D avec OpenGL, etc.

7.1 Notre formation au métier d'enseignant

Notre premier contact avec l'enseignement supérieur a été une expérience d'enseignement de la physique au niveau BAC+2 à la maison d'Arrêt de Metz-Queuleu, en 1993. Plus classiquement ensuite, nous avons durant notre thèse bénéficié du statut de Moniteur, ce qui nous a permis de prendre part à l'enseignement de l'Informatique en DEUG et Licence, sous forme de TD et TP d'introduction à la programmation. À cette occasion, nous avons pu nous intégrer dans les équipes pédagogiques concernées, et prendre part à la création des sujets. À cette formation « *sur le tas* » s'est ajoutée la structuration apportée par le CIES, c'est-à-dire le suivi de formations à l'enseignement, et l'aide précieuse d'un « *tuteur* », Jean-Pierre Jacquot en ce qui nous concerne.

7.2 Enseignement à l'Université

Bien que la plus grande partie de nos enseignements soient ceux que nous dispensons à Supélec, nous prenons part à l'enseignement universitaire suivant deux aspects. Le premier concerne l'encadrement des stages de DEA/Master informatique, qui est une formation à la recherche, et le second est l'enseignement de cours magistraux en DEA/Master Informatique.

7.2.1 Stages de DEA/Master informatique de l'UHP Nancy I

Outre la formation dont, nous l'espérons, ont bénéficié les étudiants durant les stages de Master Informatique que nous avons encadrés, l'avantage de cette activité est qu'elle permet d'alimenter ponctuellement notre projet de recherche. Plus précisément, il s'est agi de « *tester* » certaines idées, préliminaires à celles auxquelles nous donnons suite dans notre projet.

Les étudiants que nous avons encadrés étaient pour une grande partie d'entre eux (5 sur 8) des étudiants de troisième année de Supélec, et nous sommes fier d'avoir pu orienter vers le milieu académique des étudiants ayant bénéficié des filières élitistes que constituent le cursus « *École d'ingénieur* » si particulier à notre pays. Cette volonté d'orienter les ingénieurs vers l'obtention d'un doctorat est aujourd'hui partagée par Supélec, qui fait la promotion de ces parcours auprès de ses étudiants.

7.2.2 Cours au Master informatique de l'UHP Nancy I

Initialement, nous donnions avec Frédéric Alexandre un cours de DEA informatique relatif au connexionnisme, puis en 2005, au moment du passage au Master Informatique, nous avons infléchi cet enseignement pour qu'il puisse s'intégrer dans un module d'ossature. Nous intervenons aujourd'hui dans ce module avec Amédéo Napoli, et nous mettons plus l'accent sur l'apprentissage automatique numérique que sur le modèle connexionniste.

Nous souhaitons souligner ici que nous ne sommes pas partisans d'un enseignement qui serait « *recherche* » à l'Université et « *pro* » dans les écoles d'ingénieur. En effet, pour ce qui est de l'apprentissage automatique, il s'agit de concepts raffinés dont ont besoin aussi bien les ingénieurs que les chercheurs, même s'il est clair qu'il appartient aux seconds de les faire évoluer, et aux premiers de les appliquer en maîtrisant la portée. Nous avons donc, à Supélec, avec nos collègues Michel Ianotto et Olivier Pietquin, créé un enseignement de l'apprentissage automatique numérique sur deux des trois options de troisième année du campus de Metz, et cet enseignement est de même nature que celui dispensé à l'Université sur ces thèmes.

7.3 Formation technologique

L'autre grande partie des enseignements que nous effectuons sont relatifs à la formation technologique dispensée en école d'ingénieur. Pour ce qui nous concerne, nous prenons en charge les formations relatives à l'enseignement de l'algorithmique et de la programmation, à l'apprentissage automatique. Nous intervenons aussi au sein de modules d'ouvertures pour les élèves, qui dans notre cas sont une ouverture vers la biologie du système nerveux et les sciences cognitives.

7.3.1 Algorithmique et programmation

Nous avons eu un premier contact avec l'enseignement de l'algorithmique (en C++) et des systèmes informatiques en école d'ingénieur lorsque nous étions ATER à l'ESSTIN, à Nancy. Nous sommes ensuite arrivés à Supélec au moment d'une réforme de l'enseignement de tronc commun (première et deuxième année), réforme à laquelle nous avons pu prendre part. Nous avons fait, lors des discussions pédagogiques, la promotion d'une approche scientifique de l'informatique (symboles et valeurs, évaluation stricte et paresseuse, induction structurelle, etc.), que nous articulons autour de `ocaml`. Nous nous basons d'autre part sur `Java` pour l'enseignement de l'approche impérative et objet, bien que nous eussions préféré des langages comme `eiffel`. Là-encore, ces choix, alors que nous les défendons dans le cadre d'un enseignement en école d'ingénieur, sont inspirés de l'expérience d'enseignement que nous avons eue à l'Université Henri Poincaré – Nancy I.

En troisième année, nous présentons aux élèves le langage C++, en insistant sur les différences entre les problèmes qui relèvent de l'héritage et ceux qui relèvent des patrons de classe. Nous montrons également les techniques de méta-programmation, dont nous tirons parti dans nos productions logicielles. Fabrice Popineau, avec qui nous assurons ce module, présente le pendant dynamique des concepts statiques vus en C++ via le langage LISP, et l'ensemble constitue la présentation de principes de programmation avancée à nos étudiants, programmation à notre sens trop peu répandue dans l'informatique industrielle d'aujourd'hui.

7.3.2 Formation à l'apprentissage automatique

La formation à l'apprentissage automatique numérique dispensée à Supélec recouvre l'apprentissage supervisé (théorie de la généralisation, perceptrons, machines à noyaux, arbres de décision), l'apprentissage non supervisé (quantification vectorielle et analyse en composantes principales) et machines à états finis probabilistes (réseaux bayésiens, processus décisionnels de Markov, modèles graphiques). Nous assurons ce module avec deux collègues de l'équipe IMS, Michel Ianotto et Olivier Pietquin. Comme nous l'avons dit, ce module pourrait très bien être donné en module d'ossature d'un Master Recherche en Informatique, comme le prouvent les similarités entre ce programme et les enseignements auxquels nous contribuons au Master Informatique de l'Université Henri Poincaré – Nancy I.

7.3.3 Projets et stages

Les élèves de Supélec ont plusieurs occasions de réaliser des projets à portée technologique dans leur cursus. Excepté le projet informatique de deuxième année, plutôt scolaire, les autres projets (au nombre de 3) peuvent être l'occasion de contribuer au renfort dont nos activités de recherche ont besoin. De même, nous accueillons dans nos équipes des stagiaires, parfois de Supélec, mais aussi de l'Université Polytechnique de Bucarest. Dans la mesure d'une part où ces stages doivent avoir un centre de gravité technologique, contrairement aux stages de Master

dont ce n'est pas l'objectif, et dans la mesure d'autre part où nos besoins de développement sont largement du niveau de ce que l'on peut attendre en situation professionnelle, ces stages sont à l'intersection de nos besoins et des intérêts de l'étudiant. Nous renvoyons le lecteur au chapitre 9, qui est émaillé des contributions des différents stagiaires aux réalisations logicielles sur lesquelles s'appuient nos recherches.

7.4 Ouverture à la pluridisciplinarité

Supélec est conscient qu'un ingénieur doit garder l'esprit curieux et ouvert sur un ensemble de connaissances et de domaines bien plus larges que celui qui constitue le cœur de la formation de tronc commun. Ce besoin d'ouverture est accompagné dans l'organisation du cursus par l'obligation pour les élèves de choisir des modules d'ouverture parmi un panel aussi large que possible.

Nous intervenons dans deux moitiés de ces modules. L'un d'eux est le module « *infographie et synthèse d'image* » que nous animons avec Patrick Mercier. Il s'agit d'un module technologique, qui nous a permis de nous former à `OpenGL` et pour lequel nous avons développé la bibliothèque `glop` (cf. paragraphe 9.2) comme support pédagogique. Nous y abordons également le système visuel humain, ce qui nous permet de faire toucher du doigt la finesse avec laquelle la nature a su résoudre le problème technologique de la vision située.

Le pont vers nos intérêts de recherches auprès de nos élèves est encore plus marqué par le module « *modèles biologiques pour l'informatique* » que nous animons avec Frédéric Alexandre, au cours duquel nous montrons les liens entre les techniques d'apprentissage par renforcement, la quantification vectorielle, les systèmes dynamiques de type mémoire auto-assocative de Hopfield, les perceptrons multi-couches, avec le fonctionnement de différentes structures nerveuses du cerveau. L'idée est clairement d'exhiber l'intérêt que nous portons à ces sujet afin que les étudiants pour lesquels cet intérêt serait partagé puissent envisager de s'engager dans un projet de formation doctorale.

7.5 Formation Continue

De façon plus factuelle, nous avons une expérience d'enseignement dans le cadre de la formation continue que Supélec propose aux entreprises. L'un des stages que nous animons est proposée au catalogue de Supélec. Comme la majorité de nos enseignements, ce stage implique la collaboration d'un autre enseignant, en l'occurrence Laurent Bougrain, Maître de Conférences à l'Université Henri Poincaré – Nancy I. Mentionnons également une formation à l'apprentissage par renforcement que nous assurons en collaboration avec Olivier Pietquin.

La formation continue a pour nous cela d'intéressant qu'elle rappelle, dans le contexte parfois infantilisant des écoles d'ingénieurs, ce que doit être de la formation pour adultes, car c'est bien de cela que relève l'acte de formation que nous accomplissons auprès de nos élèves.

7.6 Synthèse

7.6.1 Formation à la recherche

- depuis jan. 2010* ▷ Thèse de Bassem Khouzam, Université Henri Poincaré – Nancy I, soutenance prévue fin 2012, « *Prise de décision motrice en ligne par une architecture d'inspiration corticale* ». Co-encadrement avec Frédéric Alexandre.
- depuis avril 2007* ▷ Thèse de Lucian Alecu, Université Henri Poincaré – Nancy I, soutenance prévue courant 2011, « *Structuration de traitements visuels par la boucle sensorimotrice : Une approche neuromimétique d'inspiration corticale* ». Co-encadrement avec Frédéric Alexandre.
- 2006* ▷ Thèse d'Olivier Ménard de l'Université Henri Poincaré – Nancy I, soutenue le 10 novembre 2006, « *Mécanismes d'inspiration corticale pour l'apprentissage et la représentation d'asservissements sensorimoteurs en robotique* ». Co-encadrement avec Frédéric Alexandre.
- 2007* ▷ Stage de Master d'Informatique de Jérôme Carretero, Université Henri Poincaré – Nancy I (5 mois). « *Plate-forme de comparaison de champs de neurones dynamiques* » .
- 2006* ▷ Stage de Master d'Informatique de Thomas Legrand, Université Henri Poincaré – Nancy I (5 mois). « *Apprentissage par renforcement dans un système sensorimoteur d'inspiration corticale* » .
- 2005* ▷ Stage de DEA d'Informatique de Jacques Henri, Université Henri Poincaré – Nancy I (5 mois). « *Auto-organisation visuelle corticale située : prise en compte de la rétinitopie* » .
- 2003* ▷ Stage de DEA d'Informatique de Yann-Eric Douvier, Université Henri Poincaré – Nancy I (5 mois). « *Évaluation asynchrone des éléments d'un champ de neurones corticaux* » .
- 2002* ▷ Stage de DEA d'Informatique d'Olivier Ménard, Université Henri Poincaré – Nancy I (5 mois). « *Modèle de carte corticale associative pour les asservissements sensorimoteurs en robotique* » .
- 2001* ▷ Stage de DEA d'Informatique d'Olivier Klein, Université Henri Poincaré – Nancy I (5 mois). « *Apprentissage neuronal pour la vision en robotique* » .
- 2000* ▷ Stage de DEA d'Informatique de Salma Jamoussi (co-encadrement), Université Henri Poincaré – Nancy I (5 mois). « *Implémentation d'une carte associative sur un robot autonome* » .
- 2000* ▷ Stage de DEA d'Informatique de Benoît Villaumié (co-encadrement), Université de Caen (5 mois). « *Modèle de carte corticale pour un robot doté d'une caméra mobile* » .

7.6.2 Enseignement

- De 2000 à 2005* ▷ Participation aux jurys du DEA puis Master Informatique de l'Université Henri Poincaré – Nancy I.

- depuis 2008* ▷ Cours magistral, demi-cours d'ossature « *Apprentissage* » de la spécialité « *Reconnaissance, Apprentissage, Raisonnement (RAR)* » du Master Informatique de l'Université Henri Poincaré – Nancy I. 14h par an. Apprentissage par renforcement, théorie de la généralisation, machines à vecteurs supports (SVM), perceptrons, quantification vectorielle.
- de 2005 à 2008* ▷ Cours magistral, demi-cours d'ossature « *Apprentissage* » de la spécialité « *Perception, Raisonnement, Interaction Multimodale (PRIM)* » du Master Informatique de l'Université Henri Poincaré – Nancy I. 14h par an. Apprentissage par renforcement, théorie de la généralisation, machines à vecteurs supports (SVM), perceptrons, quantification vectorielle.
- de 2000 à 2005* ▷ Cours magistral, demi-module « *Apprentissage* » de la filière « *Perception, Raisonnement et Traitement Automatique des Langues (PRTAL)* » du DEA d'Informatique de l'Université Henri Poincaré – Nancy I. 10h/an pendant 5 ans. Quantification vectorielle, Apprentissage par renforcement, Gestion du temps dans les réseaux de neurones, Neurones impulsifs.
- depuis 2002* ▷ Cours magistral annuel « *Modèles de programmation* », tronc commun 1^{ère} année de Supélec. 10h30, 3h de TD, 2 demi-journées de TP. Notions de calculabilité, programmation impérative (Java) et fonctionnelle (ocaml).
- de 2002 à 2008* ▷ TDs et TPs annuels du cours « *Fondements de l'Informatique, Structures de Données et Algorithmique* », tronc commun 1^{ère} année de Supélec. 6h de TD, 4 demi-journées de TP. Notions de complexité, de preuve de programme, de structures de données, de récursivité.
- de 2004 à 2010* ▷ Cours magistral annuel « *Modèles biologiques pour l'informatique* », Module Électif 2^{ème} année de Supélec. 9h (demi-module). Modélisation en sciences cognitives, quantification vectorielle, apprentissage par renforcement.
- depuis 2003* ▷ Cours magistral annuel « *Infographie et synthèse d'image* », Module Électif 2^{ème} année de Supélec. 9h (demi-module). Programmation OpenGL, photographie, colorimétrie, système visuel humain.
- depuis 2010* ▷ Cours magistral annuel « *Apprentissage numérique* » (demi-module) de l'option de 3^{ème} année de Supélec « *Systèmes Interactifs et Robotique (SIR)* ». 12h de cours, 2 demi-journées de TP. Théorie de la généralisation, machines à vecteurs supports (SVM), quantification vectorielle.
- depuis 2010* ▷ Travaux pratiques du tronc commun de l'option de 3^{ème} année de Supélec « *Systèmes Interactifs et Robotique (SIR)* ». Conception d'un TP avec des robots humanoïdes NAO et des caméras sous IP. 20 demi-journées.
- depuis 2010* ▷ Cours magistral optionnel « *Programmation C++ avancée* » de l'option de 3^{ème} année de Supélec « *Systèmes Interactifs et Robotique (SIR)* ». 7h30 de cours.
- depuis 2010* ▷ Cours magistral optionnel « *Vie artificielle* » de l'option de 3^{ème} année de Supélec « *Systèmes Interactifs et Robotique (SIR)* ». 3h de cours.
- de 2005 à 2010* ▷ Cours magistral annuel « *Apprentissage non supervisé* » (demi-module) de l'option de 3^{ème} année de Supélec « *Informatique et Ingénierie de la*

- Connaissance (IIC)* » . 4h30 et une demi-journée de TP. Quantification vectorielle (k-means, cartes auto-organisatrices, réseaux incrémentaux).
- de 2005 à 2010 ▷ Cours magistral annuel « *Apprentissage supervisé : Régression et Classification* » (demi-module) de l'option de 3^{ème} année de Supélec « *Informatique et Ingénierie de la Connaissance (IIC)* » . 9h et une demi-journée de TP. Théorie de la généralisation, machines à vecteurs supports (SVM), noyaux.
- de 2005 à 2010 ▷ Cours magistral annuel dans les modules du thème « *Apprentissage numérique* » de l'option de 3^{ème} année de Supélec « *Signaux, Images et Formes (SIF)* » . 4h30 et une demi-journée de TP. Théorie de la généralisation, machines à vecteurs supports (SVM), quantification vectorielle.
- de 2001 à 2005 ▷ TD « *réseaux* » sur machine (6h/an), TD « *Parallélisme* » sur machine (9h/an). option de 3^{ème} année de Supélec « *Informatique et Ingénierie de la Connaissance (IIC)* » .
- de 2000 à 2010 ▷ Cours magistral annuel « *Approches de la programmation* » (demi-module) dans l'option de 3^{ème} année de Supélec « *Informatique et Ingénierie de la Connaissance (IIC)* » . 9h et 4 demi-journées de TP. C++, héritage, généricité, interfaces graphiques.
- 2000 ▷ Cours magistral d'Informatique, semestre de transition du programme $N + i$ (étudiants étrangers) à Supélec. 18h. Architecture des machines, programmation assembleur.
- depuis 2004 ▷ Cours magistral d'Informatique annuel, semestre de transition du programme $N + i$ (étudiants étrangers) à Supélec. 9h et une demi-journée de TP (demi-module). Programmation fonctionnelle (ocaml).
- 2009 ▷ Encadrement du stage ingénieur de fin d'étude (BAC+5) de l'Université Polytechnique de Bucarest de Liviu Teris (4 mois), « *Video tracking application : A generic framework and extensions towards intelligent surveillance systems* » .
- 2009 ▷ Encadrement du stage ingénieur de fin d'étude (BAC+5) de l'Université Polytechnique de Bucarest de Sever Fundatureanu (4 mois), « *Integration of device control mechanisms with a generic client-server framework* » .
- 2006 ▷ Encadrement du stage ingénieur de fin d'étude (BAC+5) de l'Université Polytechnique de Bucarest de George Adrian Drumea (4 mois), « *Experimenting with Growing Neural Gas on a Video Surveillance Platform* » .
- 2005 ▷ Encadrement du stage ingénieur de fin d'étude (BAC+5) de l'Université Polytechnique de Bucarest de Lucian Alecu (4 mois), « *Architecture générique de Vidéo-Surveillance Multi-Capteurs* » .
- 2004 ▷ Encadrement du stage ingénieur de fin d'étude (BAC+5) de l'Université Polytechnique de Bucarest d'Anca Ghițescu (4 mois), « *Adding events management in a general-purpose neural network simulator* » .
- 2004-2005 ▷ Encadrement du stage d'Ingénieur CNAM de Jacques Weidig (12 mois), « *Réalisation d'une interface graphique d'aide à la mise au point et à la visualisation de réseaux d'automates cellulaires* » .

- 2010 et 2011 ▷ Responsable pédagogique, et enseignant (avec deux collègues), du stage de formation continue de Supélec « *Apprentissage Numérique* », formation intra-entreprise donnée à Rennes sur 3 jours, pour Technicolor.
- 2008 ▷ Participation à l'encadrement d'un stage de formation continue de Supélec sur l'apprentissage par renforcement pour EDF.
- 2000 ▷ Enseignement du C++ durant 2 jours de formation continue Supélec pour Essilor.

- depuis 2000 ▷ Encadrement de projets informatiques dans le cadre des 3 années de formation à Supélec, et de Contrats d'Études Industrielles de Supélec (formation des ingénieurs sur un projet proposé par une entreprise).

- 1999-2000 ▷ Poste d'ATER à l'ESSTIN (Nancy), enseignement de la programmation (60h TD, Bac+1), cours magistral sur le développement d'application répartie (10h de cours, 45h TD, Bac+3).
- 1996-1999 ▷ Monitorat au département Informatique de l'Université Henri Poincaré – Nancy I. DEUG MIAS, 2^{ème} année, 90h TD et 126h TP, sur 3 ans.

- 1993 ▷ Enseignement niveau DEUG à la Maison d'Arrêts de Metz-Queuleu.

Chapitre 8

Dissémination de la recherche

Nous décrivons dans ce chapitre les projets, manifestations, et publications au cours desquelles nos recherches ont eu l'occasion d'être présentées, impliquées, et discutées. Supélec, qui héberge ce projet, est une association de loi 1901, conventionnée avec l'État, ce qui lui impose un modèle économique où le personnel enseignant-chercheur doit, par ses activités, financer la moitié du budget. Il en découle la nécessité de coupler les activités de recherche avec le milieu industriel de sorte que les partenaires de l'école autres que l'état trouvent un intérêt économique à financer notre activité.

Même si la nécessité de valoriser la recherche en en permettant l'appropriation par le tissu industriel n'est pas propre à Supélec, puisqu'elle est mise en avant par exemple au niveau de l'Agence Nationale de la Recherche (ANR), elle prend, dans notre établissement un tour peut-être plus particulier. Tout d'abord, se confronter à des problèmes industriels plus appliqués en proposant des techniques de pointe est ce qu'attendent nos partenaires. Cette compétence attendue, même si elle ne relève pas de la recherche fondamentale, reste assez ardue et est, elle aussi, acérée par les acquis de longues années d'expérience. Nous souhaitons dans cette introduction rendre hommage à ceux de nos collègues qui ont un profil plus ingénieur que le nôtre, car leur action au sein de l'établissement a permis, en de nombreuses occasions, de rendre possible l'existence de notre projet de recherche.

Nous avons nous-aussi appris ce métier-là depuis 2000 où nous avons intégré les rangs de Supélec. Cette expérience nous a été d'autant plus nécessaire que c'est vers le milieu industriel que se dirigent majoritairement nos élèves. Le risque de ce contexte est toutefois le suivant : scinder notre temps de recherche en deux parties, la recherche fondamentale que nous présentons dans ce document d'une part, et des actions plus appliquées d'autre part, dont la raison d'être est plus « *alimentaire* » .

Nous pensons avoir évité cette situation schizoïde en lui faisant correspondre une autre dualité, le métier d'enseignant et le métier de chercheur. Ainsi, nous avons utilisé les temps liés à l'enseignement à Supélec pour acquérir des compétences opérationnelles en techniques de l'informatique avancée. Par exemple, l'appropriation des techniques à noyau, qui a fait l'objet de la programmation d'une bibliothèque (cf. paragraphe 9.8.1), est mobilisée dans certains des projets cités ici, mais elle est la compétence sur laquelle nous appuyons une grande partie des enseignements en apprentissage automatique que nous dispensons aussi bien à Supélec qu'au Master Informatique de l'Université Henri Poincaré – Nancy I. De même, les études que nous avons réalisées sur la quantification vectorielle, bien que cette notion soit liée au *codage tabulaire* présenté page 84, ont été motivées par la mise en place de techniques d'analyse de flux vidéo sur la plate-forme expérimentale de vidéo-surveillance que nous avons installée sur le campus de Metz.

Il n'empêche que cette activité a donné lieu à publications [Drumea and Frezza-Buet, 2007; Frezza-Buet, 2008], ce qui explique que toutes nos publications ne soient pas exclusivement centrées sur le calcul cortical.

Considérer que le caractère situé, dans le tissu industriel, de l'enseignement à Supélec doit être l'occasion d'activités contractuelles est une des conditions de viabilité du modèle économique d'une institution comme la nôtre. Bien entendu, il y a eu des moments où ces activités ont pris le pas sur la recherche fondamentale, mais il y a aussi eu des moments où ce fut le contraire.

La gestion de ce compromis se reflète dans les projets auxquels nous avons participé, car certains d'entre eux sont directement liés à notre activité de promotion du calcul cortical, alors que pour d'autres, nous arrivons en support avec les compétences d'apprentissage automatique que nous enseignons. Dans les deux cas toutefois, ces projets ont été l'occasion d'avoir l'expérience du suivi d'un projet, avec ses échéances, ses livrables, la diversité des points de vues des équipes, le souci d'exposer de façon convaincante les solutions que nous apportons. Cette expérience est générale aux différentes formes que peuvent prendre les projets de recherche collaboratifs, et nous comptons, bien entendu, y adosser la définition d'autres projets de collaborations à venir.

8.1 Participation à des projets de recherche

8.1.1 Projet Européen CLASSiC

Le projet européen CLASSiC (*Computational Learning in Adaptive Systems for Spoken Conversation*) démarre à l'heure où nous écrivons ces lignes. Il s'agit d'un projet de type STREP⁷¹ retenu dans l'appel 1 du FP7⁷² dans le domaine ICT⁷³

Le projet consiste à mettre en œuvre un système de dialogue à interface vocale, pour lequel une gestion de l'incertitude est mobilisée dans les différents composants. Cette gestion, statistique, fait intervenir des notions d'*apprentissage par renforcement* dans la gestion des états du dialogue, et d'apprentissage supervisé. Dans notre équipe, c'est Olivier Pietquin qui est spécialiste de ces questions, et nous intervenons en support pour la mise en œuvre effective des techniques d'apprentissage statistique pour lesquelles nous avons développé des logiciels dans le cadre de nos activités contractuelles avec le milieu industriel. L'ambition de ce projet est de montrer que la gestion d'un état du dialogue, global, est en mesure d'infléchir les traitements statistiques effectués dans tous les modules du système de dialogue afin de supprimer les ambiguïtés qui peuvent se manifester dans ces modules. C'est un exemple de restitution de cohérence, même si pour ce projet-là, ce ne sont pas les techniques de calcul cortical qui devraient a priori être mises en avant.

Plus précisément, notre contribution au projet a permis de jeter les fondements d'une bibliothèque C++ implémentant des techniques d'apprentissage par renforcement, autorisant des méthodes de régression de la fonction de valeur pour traiter les espaces d'état continu. Ces travaux font l'objet d'une collaboration avec Matthieu Geist et Olivier Pietquin au sein de notre équipe [Pietquin *et al.*, 2011].

8.1.2 Projet CPER : InterCell

Depuis fin 2007, nous sommes coordinateur du projet *InterCell*. Ce projet a fait l'objet d'un financement en matériel de 450 Keuros, dans le cadre du premier appel à projet de l'axe

71. Specific Targeted Research Projects.

72. Framework Programme

73. Information and Communication Technologies.

MIS⁷⁴ du PRST MISN⁷⁵ du contrat de plan État-Région Lorraine 2007-2013. La durée du projet est de 3 ans, et à l'heure où nous écrivons ces lignes, le cluster, avec la logistique qui lui est afférente⁷⁶, est installé sur le campus de Metz de Supélec.

Le cœur du projet est le financement d'un cluster de PC, de 256 machines *dual core* à 2.66 GHz, disposant de 4 Go de mémoire chacune. Ce qui fait l'originalité de cet investissement est qu'il est destiné à promouvoir le *calcul à grain fin*, ce qui est un défi sur ces machines, mais également un calcul qui soit *interactif*. La stratégie que nous avons adoptée pour la proposition de ce sujet est la suivante.

D'une part, les physiciens sont demandeurs de calcul intensif pour réaliser des simulations numériques, et leur fournir cette puissance peut nous amener à prendre part à des projets comme ITER, ou à mettre un pied dans les nano-technologies. Ces domaines, bien qu'ils soient des domaines de recherche pour la physique, sont pour nous, informaticiens, des applications de nos compétences.

D'autre part, la parallélisation sur cluster de systèmes à grain fin est également une problématique en soi en Informatique [Vialle, 2002]. Cette problématique en rencontre une autre, la nôtre, à savoir la promotion en Informatique du calcul à grain fin. Cette dernière est plus amont, en ce sens qu'aujourd'hui, le calcul à grain fin n'est pas encore une technique reine pour résoudre des problèmes industriels aux enjeux économiques court-terme.

Le projet *InterCell* donne une cohérence aux besoins relatifs au calcul intensif, et à des problématiques de recherche plus fondamentales en Informatique. Le trait d'union est ici assuré par la démonstration de la faisabilité de la résolution d'un système d'équations aux dérivées partielles par le calcul à grain fin. Nous avons établi cette faisabilité avec Nicolas Fressengeas, Professeur de Physique au Laboratoire LMOPS à Metz, en concevant et réalisant le simulateur *ESCAPaDE* (cf. 6.1).

Le calcul à grain fin se retrouve alors au cœur de problématiques informatiques de recherche fondamentale d'une part, au niveau du parallélisme et au niveau du calcul cortical, mais aussi au cœur des enjeux économiques afférents aux simulations numériques en physique, où il intervient alors plus comme un outil de calcul que comme un objet de recherche. La notion d'interactivité s'ajoute alors dans un cas comme dans l'autre. En informatique, l'interactivité prend son sens à deux titres. Le premier est celui de la visualisation de l'évolution de ces systèmes, qui sont, comme nous l'avons dit au paragraphe 2.1.1, difficiles à appréhender. Le second concerne l'approche située, puisqu'un calculateur à grain fin de ce type ne peut s'embarquer sur un robot, mais doit interagir en permanence avec ses capteurs et ses effecteurs. Pour la physique, l'interactivité autorise la visualisation d'une convergence en cours, ce qui permet d'affiner la modélisation. Parfois en effet, du fait de problèmes de normalisation des équations, une des équations du système prend le pas sur les autres, ce qui est bien plus facile à voir au début de la convergence qu'à partir d'un fichier reçu en fin de simulation, du type de ceux que l'on reçoit lorsque l'on utilise un cluster plus classique. L'interaction est dans ce cas une aide à la constitution et à l'ajustement des modèles proposés par les physiciens. Dans ce projet, les partenaires suivants sont impliqués. Nous listons ces partenaires en identifiant leur niveau d'implication, au regard de la cohérence et de la complémentarité que nous avons décrites dans ce qui précède.

- Partenaires informaticiens :
 - Équipe IMS (Supélec, campus de Metz) : Parallélisation du calcul à grain fin, calcul cortical situé.

74. Modélisation, Interaction, Simulation

75. Modélisations, simulations et systèmes numériques

76. climatisation, aménagement de salle, acheminement des fluides, etc.

- Équipe Algorille (Loria, Nancy) : Parallélisation du calcul à grain fin.
- Équipe Cortex (Loria, Nancy) : Calcul cortical situé.
- Partenaires physiciens :
 - Laboratoire LMOPS (Metz) : Conception du simulateur d'équations différentielles, simulations d'équations aux dérivées partielles pour l'optique.
 - Équipe « *plasmas chauds* » (LMPIA, Nancy) : simulations d'équations aux dérivées partielles pour l'étude des plasmas.
 - UMI-2958 : simulations d'équations aux dérivées partielles pour l'optimisation des caractéristiques et de la propagation de lumière dans les nano-structures.
- Partenaire mathématicien :
 - Équipe CALVI (INRIA, Nancy-Strasbourg) : Modélisation des plasmas.

Nous renvoyons le lecteur désireux de plus d'informations à [InterCell, 2008], ce site étant tenu à jour durant le déroulement du projet. Nous nous contenterons ici de souligner que le cluster est aujourd'hui opérationnel, d'un point de vue matériel comme logiciel (cf. paragraphe 9.7), et que sa particularité est utilisé comme argument dans le montage de projets collaboratifs (en cours).

8.1.3 Projet Européen MirrorBot

Nous sommes intervenu en tant que sous-traitant dans le projet européen **MirrorBot**, pour l'encadrement de la thèse d'Olivier Ménard, cofinancée par cette sous-traitance et la Région Lorraine.

MirrorBot a été un projet européen FET⁷⁷/IST⁷⁸ s'étalant sur 3 ans, cherchant à instancier sur plate-forme robotique le concept de *neurones miroir* avancé par l'équipe de Giacomo Rizzolatti, partenaire du projet.

Nous avons apporté à ce projet nos travaux concernant le codage visuel, présentés au paragraphe 9.6.4, ainsi que l'étude de modules corticaux pour les représentations multimodales présentée au paragraphe 4.5. Les partenaires du projet étaient :

- L'équipe du Professeur Stefan Wermter, Chair for Intelligent Systems, School of Computing and Technology, University of Sunderland. Robotique et réseaux de neurones, apprentissage par renforcement.
- L'équipe du Professeur Günther Palm, Neuroinformatics, Université de Ulm. Modélisation bio-inspirées par des neurones impulsionnels.
- L'équipe Cortex, Dirigée par Frédéric Alexandre. Calcul cortical pour la multimodalité. C'est dans ce cadre que nous sommes intervenus dans le projet.
- L'équipe du Professeur Giacomo Rizzolatti, Institute of Neurophysiology, Université de Parme. Étude des neurones miroir chez le singe.
- L'équipe du Professeur Friedemann Pulvermüller, Medical Research Council, Cognition and Brain Sciences Unit. Étude du codage cortical du langage, par IRMf notamment.

Pour ce qui nous concerne, ce projet a été l'occasion, avec Olivier Ménard, de concevoir le modèle **bijama** et de le soumettre au point de vue des partenaires biologistes, en particulier Victorio Gallese, de l'équipe de Giacomo Rizzolatti, ce qui fut extrêmement enrichissant d'un point de vue scientifique.

77. Future and Emerging Technologies

78. Information Society Technologies

8.1.4 Projet Avim du programme Robea

Le programme Robea est un programme interdisciplinaire du CNRS qui a financé des actions de collaboration entre équipes de recherches sur le thème de la robotique. Ce programme, dirigé par Malik Galhab (DR CNRS, LAAS), a été par son ouverture à d'autres disciplines que l'automatique précurseur de la thématique « *robotique cognitive* » présente aujourd'hui au sein de l'ANR.

En 2002, nous avons assumé la coordination d'un projet au sein de ce programme, le projet « *Apprentissage de transformations visio-motrices (AVIM)* », d'un budget, sur 2 ans, de 150 Keuros, dont 50 étaient financés par Robea. Nous avons, entre autres, financé une partie de notre plate-forme robotique avec ce projet (cf. figure 8.1).

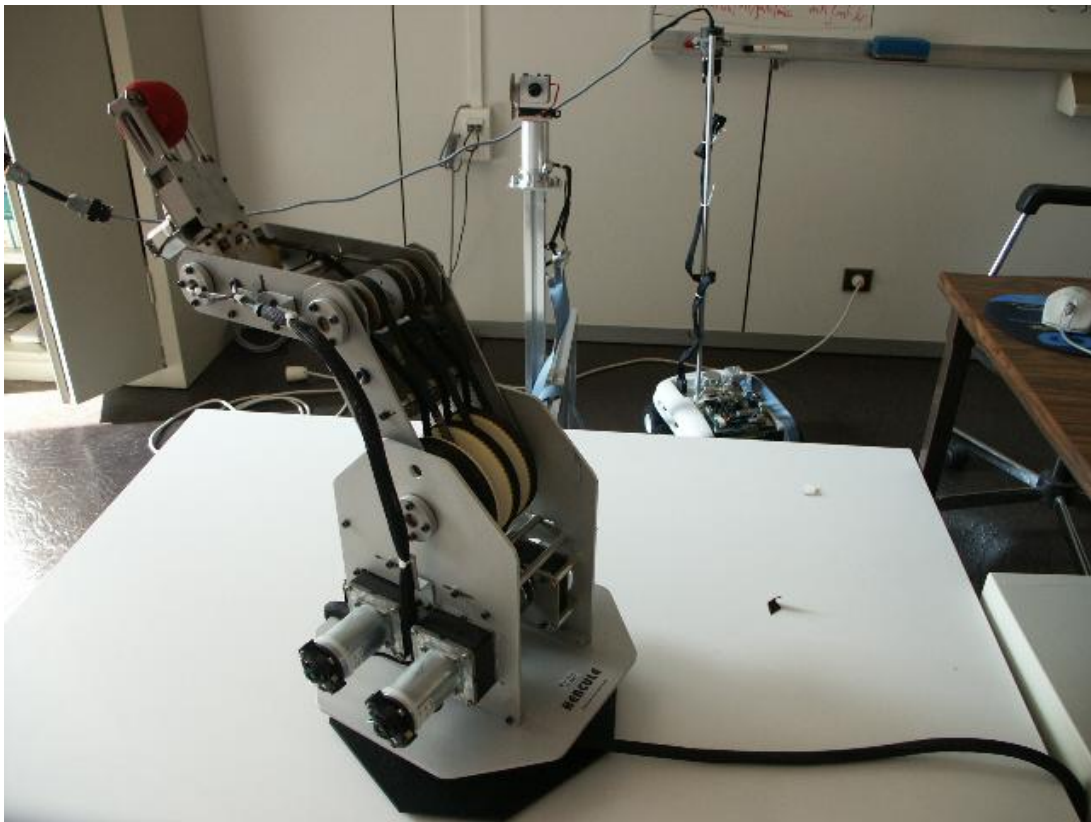


FIGURE 8.1 – Plate-forme robotique pour l'asservissement main-œil. Le bras articulé a été financé par le projet Avim du programme Robea.

L'objectif scientifique du projet était de comprendre comment l'on pouvait, par des modules indifférenciés implémentant un calcul cortical, prendre en charge les différents degrés de liberté d'un bras articulé. C'est au cours de ce projet que nous avons confronté deux approches, l'approche par *codage tabulaire* et l'approche par « *codage monotone* » pour la prise en compte de la proprioception, approches a priori antagonistes par les partenaires du projet. Nous en sommes arrivés à la conclusion que le caractère tabulaire des représentations peut être concilié avec les propriétés d'extrapolation observées expérimentalement chez les primates si l'on part du principe que le cortex commande des successions de postures, plutôt que directement chacune des articulations. L'application à la télé-opération était soutenue par EDF, dont la problématique était de réparer des canalisations en y introduisant des robots manipulateurs, et qui souhaitait

étudier la possibilité d'en assister la commande par des modèles inspirés de la biologie.

Ce projet n'a pu toutefois être finalisé autant que nous l'aurions souhaité, du fait de la décision des participants dépendant du CNRS, en fin de projet, de geler leur participation, consécutivement aux problèmes budgétaires du CNRS à cette époque.

Les partenaires étaient les suivants :

- Équipe ERSISP, Supélec, campus de Metz (avant l'existence d'IMS). Calcul cortical sur plate-forme robotique.
- Équipe Cortex, Loria, Nancy. Architectures multimodales d'inspiration corticale.
- Équipe « *Plasticité cérébrale et adaptation des fonctions visuelles et motrices* », INSERM U483, Paris. Codage monotone et expérimentation.
- Équipe « *Optimisation des Chantiers de Maintenance* », EDF, Chatou. Télé-opération à l'aide de bras articulés.

8.1.5 Projet « *Télé-assistance mobile* » du CPER-Lorraine

Dès notre arrivée à Supélec, en mars 2000, nous avons participé à l'action « *Télé-assistance mobile* » du projet « *Téléopérations et assistants intelligents (TOAI)* » du PRST⁷⁹ « *Intelligence Logicielle* » du CPER-Lorraine. Au niveau financier, ce projet nous a permis d'acquérir une partie de notre plate-forme robotique (2 robots *koala*, dont un est visible sur la figure 8.1, et trois caméras pan-tilt-zoom *axis PTZ 213*).

Le but du projet était la mise en place de capteurs intelligents, et dans notre cas de capteurs mobiles. Dans notre équipe, alors animée par Stéphane Vialle, des travaux ont été menés concernant la robustesse d'applications redondantes, s'exécutant sur plusieurs sites mais concernant la même machine physique. Nous renvoyons à [Vialle, 2002] pour une présentation de ces travaux, ne retenant pour ce qui concerne notre projet de recherche que cette action, qui s'est déroulée dans les premiers temps de notre arrivée à Supélec, nous a permis de mettre en place la plate-forme logicielle avec laquelle nous gérons, aujourd'hui encore, notre matériel robotique et la *smartroom* (cf. paragraphe 9.9).

8.2 Actions d'animation de la recherche

8.2.1 Organisation de conférences

JFPDA'08

Nous avons participé, avec Olivier Pietquin, à l'organisation sur le campus de Metz de Supélec des « *Journées Francophones Planification, Décision, Apprentissage pour la conduite de systèmes* », qui se sont tenues les 19 et 20 Juin 2008 [JFPDA, 2008].

Neurocomp'06

Nous étions membre du comité d'organisation de la première édition de la Conférence en Neurosciences Computationnelles (Neurocomp), qui s'est tenue à Pont-à-Mousson, les 23-24 octobre 2006 [NeuroComp, 2006]. Cette conférence permet de réunir la communauté française de ce domaine, mais invite également des conférenciers étrangers. Nous nous sommes également rendu à l'édition suivante, organisée à l'ESPCI à Paris, les 14-16 Novembre 2007.

79. Pôle de Recherche Scientifique et Technologique

8.2.2 Autres actions

Réseau Grand-Est des Sciences Cognitives

Le réseau Grand-Est des Sciences Cognitives a initialement accompagné L'ACI Cognitive du MENRT (1999-2003), mais reste aujourd'hui un réseau d'équipes, concernées à divers titres par les problématiques des sciences cognitives, réseau auquel nous appartenons depuis 2000. Le réseau finance l'organisation de séminaires thématiques, organisation à laquelle nous avons pris part à deux reprises.

Les 23 et 24 mai 2002, nous avons organisé sur le campus de Metz des « *Journées du réseau Grand-Est des Sciences Cognitives* », sur le thème des techniques d'apprentissage automatique issues de la biologie (quantification vectorielle, apprentissage par renforcement, etc.). Nous avons également organisé une après-midi des « *Ateliers scientifiques sur l'émotion et la mémoire spatiale* » qui se sont tenus à l'Université Louis Pasteur, à Strasbourg, le 8 Juin 2006.

Suppléance perceptive

Nous avons, en 2003 et 2004, participé aux réunions du Projet d'Action Spécifique « *Suppléance perceptive et interface* », coordonné par Olivier Gapenne et Philippe Gaussier. Ces réunions ont été l'occasion d'échanges avec des chercheurs de disciplines qui abordent les sciences cognitives sous un angle psychologique et philosophique. Nous avons pu au cours de ces réunions rencontrer, entre autres, des chercheurs comme Kevin O'Regan et Alva Noë, que nous citons abondamment dans ce mémoire.

De façon plus anecdotique, nous avons également eu l'occasion, lors de ces journées, de participer à un atelier où nous avons taillé du silex, non sans mal il faut bien le reconnaître, à la manière des hommes préhistoriques. Nous en avons retenu quelques bleus, beaucoup de déchet, une petite lame de silex miraculeusement obtenue à la suite de percussions tâtonnantes, que nous gardons précieusement, et surtout un regard différent sur les bifaces que l'on peut observer dans les musées de la préhistoire.

TopVision

Depuis 2005, nous participons, avec d'autres laboratoires, à la confrontation et la comparaison d'algorithmes d'apprentissage et de traitement du signal sur une plate-forme commune, fournissant des films sous-marins sur lesquels il s'agit de détecter des objets. Le projet TOPVISION est coordonné par THALES Underwater Systems SAS et rattaché au programme Techno-Vision lancé par le Ministère de la Recherche français et le Ministère de la Défense français par l'intermédiaire du GESMA (Groupe d'Etudes Sous-Marines de l'Atlantique).

Nous avons présenté au cours de ce programme nos travaux relatifs au suivi de distribution non-stationnaire par quantification vectorielle (cf. paragraphe 6.2).

Amina

Nous avons participé aux trois conférences « *Applications Médicales de l'Intelligence Neuro-Artificielle (AMINA)* », organisées à la Faculté de Médecine de Monastir. L'objet des interventions que nous y avons faites était de présenter aux médecins les techniques d'apprentissage automatique. Lors de la première conférence, en 2000, nous avons présenté la problématique de la robotique autonome par des méthodes inspirées du cortex. En 2004, lors de la deuxième conférence, nous avons fait une présentation des algorithmes génétiques, et avons animé des travaux

pratiques sur l'apprentissage artificiel. En 2006 enfin, nous avons présenté les enjeux du calcul cortical, tel que nous le concevons avec le modèle `bijama`.

Nous souhaitons ici saluer les partenaires tunisiens, car ils savent, par la qualité de leur accueil, ajouter à l'intérêt scientifique de ces rencontres une chaleur très méditerranéenne, qui fait de ces conférences des occasions de rencontres pluridisciplinaires des plus agréables.

8.3 Publications

MÉMOIRE DE DOCTORAT

- ▷ Hervé Frezza-Buet. *Un modèle de cortex pour le comportement motivé d'un agent neurorimimétique autonome*. PhD thesis, Université Henri Poincaré - Nancy I, oct 1999. In french.

ARTICLES DE JOURNAUX

- ▷ Olivier Pietquin, Matthieu Geist, Senthilkumar Chandramohan, and Hervé Frezza-Buet. Sample-Efficient Batch Reinforcement Learning for Dialogue Management Optimization. *ACM Transactions on Speech and Language Processing*, 7(3), May 2011. 21 pages.
- ▷ Lucian Alecu, Hervé Frezza-Buet, and Frédéric Alexandre. Can self-organization emerge through dynamic neural fields computation? . *Connection Science*, 23(1) :1–31, 2011.
- ▷ Hervé Frezza-Buet. Following non-stationary distributions by controlling the vector quantization accuracy of a growing neural gas network. *Neurocomputing*, 71(7-9) :1191–1202, 2008. extended version of *Tracking fast changing non-stationary distributions with a topologically adaptive neural network : application to video tracking*, ESANN'07.
- ▷ Olivier Ménard and Hervé Frezza-Buet. Model of multi-modal cortical processing : Coherent learning in self-organizing modules. *Neural Networks*, 18(5-6) :646–655, 2005. extended version of *Coherent learning in cortical maps : A generic approach*, IJCNN'05.
- ▷ Hervé Frezza-Buet and Frédéric Alexandre. From a biological to a computational model for the autonomous behavior of an animat. *Information Sciences*, 144(1-4) :1–43, 2002.

CHAPITRES DE LIVRES

- ▷ Frédéric Alexandre and Hervé Frezza-Buet. From natural to artificial intelligence : Numerical processing for cognitive tasks. In Jean-Pierre Nadal Paul Bourguine, editor, *Cognitive Economics : An Interdisciplinary Approach*, pages 113–130. Springer, may 2004.
- ▷ Hervé Frezza-Buet and Frédéric Alexandre. Learning selection of action for a cortically-inspired robot control. In John Demiris and Andreas Birk, editors, *Interdisciplinary Approaches to Robot Learning*, Robotics and Intelligence Systems Series, pages 95–113. World Scientific Publishers, 1999.

ARTICLES LNCS

- ▷ Lucian Alecu and Hervé Frezza-Buet. Application-driven parameter tuning methodology for dynamic neural field equations. In *Neural Information Processing, ICONIP'09 Proceedings, Part I*, volume 5863/2009 of *Lecture Notes in Computer Science*, pages 135–142, Bangkok (Thailand), 2009. Springer Berlin / Heidelberg.
- ▷ Hervé Frezza-Buet and Olivier Ménard. Making competition in neural fields suitable for computational architectures. In Wlodzislaw Duch, Janusz Kacprzyk, and Erkki Oja, editors, *International Conference on Artificial Neural Networks (ICANN)*, Lecture Notes in Computer Science (LNCS) 3696, part I, pages 217–222. Springer-Verlag, 2005.
- ▷ Olivier Ménard, Frédéric Alexandre, and Hervé Frezza-Buet. Towards word semantics from multi-modal acoustico-motor integration : Application of the bijama model to the setting of action-dependant phonetic representations. In *Biomimetic Neural Learning for Intelligent Robots : Intelligent Systems, Cognitive Robotics, and Neuroscience*, Lecture Notes in Computer Science (LNCS) 3575, chapter 8, pages 144–161. Springer, jul 2005.
- ▷ Hervé Frezza-Buet. Action scheme scheduling with a neural architecture : A prefrontal cortex approach. In J.R. Dorransoro, editor, *International Conference on Artificial Neural Networks (ICANN)*, Lecture Notes in Computer Science (LNCS) 2415, pages 271–276. Springer-Verlag, 2002.
- ▷ Hervé Frezza-Buet, Nicolas Rougier, and Frédéric Alexandre. Integration of biologically inspired temporal mechanisms into a cortical framework for sequence processing. In Ron Sun and Lee C. Giles, editors, *Sequence Learning : Paradigms, Algorithms, and Applications*, Lecture Notes in Computer Science (LNCS) 1828, pages 321–348. Springer-Verlag, 2001.

ARTICLES DE CONFÉRENCES INTERNATIONALES

- ▷ Jens Gustedt, Stephane Vialle, Hervé Frezza-Buet, D'havh Boumba Sitou, and Nicolas Fressengeas. InterCell : a Software Suite for Rapid Prototyping and Parallel Execution of Fine Grained Applications. In *PARA 2010 conference : State of the Art in Scientific and Parallel Computing*, Reykjavick, Iceland, june 6-9 2010.
- ▷ Jean-Louis Gutzwiller, Hervé Frezza-Buet, and Olivier Pietquin. Online Speaker Diarization with a Size-Monitored Growing Neural Gas Algorithm. In Michel Verleysen, editor, *Proceedings of the 18 th European Symposium on Artificial Neural Networks (ESANN), Computational Intelligence and Machine Learning*, pages 505–510, Bruges (Belgium), April 2010.
- ▷ Lucian Alecu and Hervé Frezza-Buet. Reconciling neural fields to self-organization. In Michel Verleysen, editor, *European Symposium on Artificial Neural Networks - Advances in Computational Intelligence and Learning (ESANN), Bruges, Belgium*, pages 571–576, April 2009.
- ▷ Lucian Alecu and Hervé Frezza-Buet. Are neural fields suitable for vector quantization ? In *Proceedings of The Seventh International Conference on Machine Learning and Applications (ICMLA'08)*, pages 239–244, San Diego (CA, USA), December 2008. IEEE.
- ▷ Nicolas Fressengeas, Hervé Frezza-Buet, Jens Gustedt, and Stephane Vialle. An interactive problem modeller and pde solver, distributed on large scale architectures. In *Third International Workshop on Distributed Frameworks for Multimedia Applications (DFMA-07)*, 2007. 8 pages.

- ▷ Georges Adrian Drumea and Hervé Frezza-Buet. Tracking fast changing non-stationary distributions with a topologically adaptive neural network : application to video tracking. In *ESANN, European Symposium on Artificial Neural Networks*, 2007. 6 pages.
- ▷ Olivier Ménard and Hervé Frezza-Buet. Coherent learning in cortical maps : A generic approach. In *IEEE International Joint Conference on Neural Networks (IJCNN)*, volume 5, pages 2885–2890, 2005.
- ▷ Frédéric Alexandre, Hervé Frezza-Buet, and Olivier Ménard. Multi-criteria self-organization : Example of motor-dependant phonetic representation for a multi-modal robot. In *AI workshop on Neurobotics, 27th Conference on Artificial Intelligence (KI-2004)*, Ulm, Germany, sep 2004.
- ▷ Stephane Vialle, Olivier Ménard, and Hervé Frezza-Buet. Making cortically-inspired sensorimotor control realistic for robotics : Design of an extended parallel cellular programming model. In *International Conference on Advances in Intelligent Systems - Theory and Applications. In cooperation with the IEEE Computer Society*, nov 2004.
- ▷ Olivier Ménard and Hervé Frezza-Buet. Rewarded multi-modal neuronal self-organization : Example of the arm reaching movement. In *International Conference on Advances in Intelligent Systems - Theory and Applications. In cooperation with the IEEE Computer Society*, nov 2004.
- ▷ Olivier Ménard and Hervé Frezza-Buet. Multi-map self-organization for sensorimotor learning : a cortical approach. In *IEEE International Joint Conference on Neural Networks*, volume 4, pages 3054–3059, 2003.
- ▷ Hervé Frezza-Buet and Frédéric Alexandre. Specialization within cortical models : An application to causality learning. In *Proceedings of the 7th European Symposium on Artificial Neural Networks (ESANN)*, 1999. 6 pages.
- ▷ Hervé Frezza-Buet and Frédéric Alexandre. Modeling prefrontal functions for robot navigation. In *IEEE International Joint Conference on Neural Networks (IJCNN)*, pages 252–257, 1999.
- ▷ Nicolas Rougier, Hervé Frezza-Buet, and Frédéric Alexandre. Neuronal mechanisms for sequence learning in behavioral modeling. In C L Giles and R Sun, editors, *Reinforcement Methods for Sequence Learning Workshop, Sixteenth International Joint Conference on Artificial Intelligence (IJCNN)*, pages 35–40, 1999.
- ▷ Hervé Frezza-Buet and Frédéric Alexandre. Selection of action with a cortically-inspired model. In *Seventh European Workshop on Learning Robots (EWLR)*, pages 13–21, 1998.
- ▷ Hervé Frezza-Buet and Frédéric Alexandre. Multimodal sequence learning with a cortically-inspired model. In Paul Wang, editor, *Joint Conference on Information Sciences (JCIS), Association for Intelligent Machinery*, volume 2, pages 24–27, 1998.
- ▷ Hervé Frezza-Buet and Frédéric Alexandre. A model of cortical activation for robot navigation. In *Second International Conference on Cognitive and Neural Systems*, 1998. poster presentation.

AUTRES

- ▷ Lucian Alecu and Hervé Frezza-Buet. A dynamic neural field mechanism for self-organization. In *Supplement of the online journal BMC Neuroscience, The nineteenth Annual Computational Neuroscience Meeting, Berlin, Germany*, 2009.

- ▷ Lucian Alecu and Hervé Frezza-Buet. An empirical evaluation framework for qualifying dynamic neural fields. In Laurent E. Perrinet and Emmanuel Daucé, editors, *Proceedings of the second french conference on Computational Neuroscience, Neurocomp, Marseille*, page (4 pages), October 2008.
- ▷ *Actes des Journées Francophones de Planification, Décision et Apprentissage pour la conduite de systèmes (JFPDA 2008)*, June 2008.

Chapitre 9

Production logicielle

Le projet de recherche que nous présentons dans ce document fait appel à des ressources logicielles diverses, dont nous avons profité. Nous rendons ici hommage au monde du Logiciel Libre, qui est le contributeur exclusif des logiciels que nous avons utilisés. En retour, nous nous sommes efforcé de fournir nous-aussi des logiciels libres, dont le code est ouvert. Ce sont ces logiciels qui sont brièvement présentés ici.

9.1 Politique de développement logiciel

Dans un contexte de recherche, la production de logiciels n'est pas ce qui est le plus valorisé. En effet, une fois un résultat établi, un algorithme conçu, la conception du logiciel associé ne relève pas de la recherche à proprement parler, et n'aborde pas de questions scientifiques particulières.

Toutefois, il est essentiel pour toute équipe de recherche de pouvoir gérer correctement la production logicielle qui découle de ses recherches, sous peine de ne pouvoir exploiter les travaux passés, de doctorants par exemple. Or la pérennisation de code suppose qu'il y ait des ingénieurs pour le maintenir, le documenter, le distribuer, ce qui n'a pas été le cas dans notre équipe. À ceci s'ajoute la création récente de la plate-forme *smartroom* sur le campus de Metz de Supélec (cf. paragraphe 9.9), qui a eu un impact fort sur notre développement logiciel, dans la mesure où le matériel associé à cette plate-forme doit être pour nous opérationnel, aussi bien pour y expérimenter notre recherche que pour l'utiliser dans le cadre de la formation des étudiants de Supélec (projets, travaux pratiques de l'option de troisième année « *Systèmes Interactifs et Robotique* » (SIR)).

Face à la nécessité d'avoir une assise logicielle saine à nos travaux, avec un coût de maintenance réduit, nous avons dû trouver une solution compatible avec le fait que notre activité n'ait pas pour finalité la production logicielle, mais la production scientifique.

Cette solution consiste premièrement à fixer le système et l'environnement de programmation. Nous avons retenu le langage C++ et le système Unix (Linux en l'occurrence). Le langage C++ a pour nous l'intérêt de permettre une approche objet de haut niveau, par l'usage en particulier des patrons de classe, tout en assurant une vraie proximité avec la machine, qui est un gage d'efficacité.

Deuxièmement, nous avons rendu le code le moins monolithique possible. Par exemple, la prise en charge des communications réseaux est assurée par une seule bibliothèque, que ce soit dans le cadre de la plate-forme de video-surveillance de Supélec ou dans celui de la visualisation de réseaux de neurones. Ainsi, chaque logiciel profite immédiatement de l'évolution des logiciels

dont il dépend. Cette philosophie d'une production logicielle « *éclatée* », empruntée au monde Unix, est de fournir un ensemble de logiciels élémentaires qui font une seule chose, mais qui la font bien, de façon ouverte à l'interfaçage avec d'autres éléments.

Troisièmement, un code n'est utilisable de façon pérenne que s'il est documenté, ou du moins si son interface l'est. Il est impossible dans notre environnement de travail de maintenir une documentation à jour avec les évolutions du code. Nous avons donc dû trouver un compromis, en assurant que le code soit sa propre documentation. Ainsi, nous faisons un usage systématique de *Doxygen* pour générer un manuel de référence à partir de commentaires adéquats dans les fichiers `.h`, et toutes nos bibliothèques sont fournies avec des exemples d'utilisation, qui nécessairement « *compilent* », faisant office de manuel d'utilisation.

Quatrièmement, nous avons choisi de distribuer notre code sous licence gratuite, principalement la licence Lesser-GPL, mais parfois aussi la licence Free-BSD. La raison est que ces licences permettent l'utilisation de notre code à des fins commerciales, ce qui est compatible avec les missions de Supélec vis-à-vis de ses partenaires industriels. Nous distribuons nos logiciels sous forme de *packages* source, les fameux fichiers `.tar.gz` impliquant les outils GNU `automake`, `autoconf`, `libtool`, `pkg-config` et `gcc`. Ces outils sont essentiels pour la mise en cohérence d'une production logicielle « *éclatée* ». Dans la mesure où c'est la distribution Fedora de Linux qui est maintenue sur le campus, nous fournissons également un support d'installation de *packages* binaires, qui sont des fichiers `.rpm`, via le logiciel `yum`. La façon dont sont conçus nos logiciels source, du fait de l'utilisation des outils GNU, permettrait un *packaging* compatible avec d'autres distributions, mais nous ne l'avons pas fait pour l'instant.

Pour résumer, prenons l'exemple d'une bibliothèque `foo`, et montrons comment nous la mettons à disposition de la communauté, d'après ce que nous avons décrit dans ce paragraphe. Tous nos logiciels s'utilisent comme nous le décrivons pour cet exemple.

Pour l'installation de `foo` sur une machine unix à partir des sources, il faut récupérer sur notre site `foo-1.00.tar.gz` puis effectuer en tant que super-utilisateur (`root`) :

```
tar zxvf foo-1.00.tar.gz
cd foo-1.00
./configure --prefix=/usr
make install
```

À partir des *packages* binaires que nous fournissons, sous la distribution Fedora, on réalise la même chose en effectuant tout simplement :

```
yum install foo-devel
```

Cette méthode a le bon goût d'installer également tous les logiciels dont `foo` dépend, et ce automatiquement.

Pour utiliser la bibliothèque `foo`, on se reportera à la documentation générée par *Doxygen*, disponible sur le site, qui fournit la description des APIs et un jeu d'exemples didactiques. La compilation de code impliquant la bibliothèque se fait alors comme suit :

```
g++ -c 'pkg-config --cflags foo' source-1.cc
g++ -c 'pkg-config --cflags foo' source-2.cc
...
g++ -c 'pkg-config --cflags foo' source-n.cc
g++ -o executable source-*.o 'pkg-config --libs foo'
```

La distribution et l'utilisation de nos logiciels répond à ce schéma qui est aujourd'hui standard dans le monde du logiciel libre, ce qui évite des problèmes de `makefile` et de gestion de variables d'environnement qui s'avèrent souvent réhébitoraires.

Cette discipline dans la production logicielle a été pour nous une question de survie, et s'est avérée payante puisque nous fournissons aujourd'hui une quarantaine de *packages*, qui s'articulent tous harmonieusement. Nous avons conçu un didacticiel permettant de construire ce type de *package*, de sorte que nos collaborateurs (les doctorants et stagiaires) puissent développer de cette façon.

Signalons enfin que l'on peut également créer des *packages* vides dont le rôle est simplement d'exprimer des dépendances. C'est le cas des *packages smartroom* et *escapade* par exemple, qui permettent en invoquant leur installation par une seule commande `yum` d'installer les suites logicielles correspondantes.

Nous mettons en avant, dans les paragraphes suivants, les principales réalisations logicielles que nous proposons, classées par thème.

9.2 Visualisation 3D

La visualisation 3D est un élément important à plusieurs titres. Dans le cas des champs de neurones dynamiques, elle permet d'appréhender intuitivement l'évolution du système (cf. figure 4.15 par exemple). Cette visualisation a également un « *effet démo* » non négligeable, lorsqu'il s'agit de promouvoir nos recherches.

Pour réaliser cette visualisation, nous avons ajouté au dessus d'`openGL` une interface objet, permettant de manipuler aisément des objets dans une scène, et une caméra. La bibliothèque `glop`⁸⁰ que nous avons développée, et à laquelle Jacques Weidig a contribué [Weidig, 2005], est également un outil pédagogique pour l'enseignement d'`openGL` que nous dispensons à Supélec.

9.3 Traitement d'images

Nous avons constaté que nos collègues traiteurs de signal ainsi que nos élèves étaient prompt à développer des traitements d'images à l'aide de fonctions intégrées de `MatLab`, mais qu'il était assez difficile de les amener à programmer ces mêmes traitements en `C++`, dont l'exécution est bien plus efficace. Non satisfaits des interfaces proposées par d'autres bibliothèques, nous avons, avec notre collègue Frédéric Pennerath et Xavier Prévost, élève de Supélec qui a effectué son stage de troisième année dans notre équipe, développé une bibliothèque `C++` de traitement d'images. La conception est basée sur les patrons de classes et la méta-programmation, ce qui permet de faire générer par le pré-compilateur, à partir d'un code d'aussi haut niveau que les fonctionnalités de `MatLab`, un code optimisé comme les codes illisibles de manipulation de tableaux dont les programmeurs `C` sont étonnamment friands.

Cette bibliothèque, appelée `mirage`, est en extension permanente, se nourrissant des contributions des collègues de l'équipe IMS. Elle contient aujourd'hui des opérateurs de filtrage, la gestion de débordement des images, le redimensionnement avec interpolation bicubique, des opérateurs morpho-mathématiques, des transformées de Hough, etc. Outre son implication dans des projets plus sérieux, nous promovons dans l'équipe une démonstration de traitement vidéo en temps réel consistant à transformer un simple manche à balai peint en vert en un sublime sabre laser, digne des plus respectables jedis (cf. figure 9.1).

80. `GL` Objects can be Pretty.

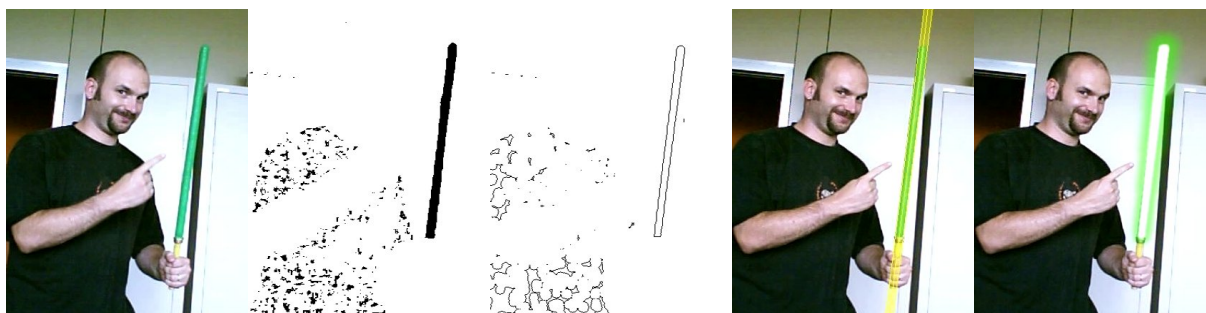


FIGURE 9.1 – Transformation en temps réel de votre serveur en justicier de l’espace. Le film est disponible sur <http://ims.supelec.fr>. On applique avec *mirage* une détection de couleur, une extraction de contours avec des filtres morpho-mathématiques, une transformée de Hough pour détecter les bords du sabre, et des opérations gaussiennes pour l’effet de halo.

9.4 Enseignement

Dans le cadre de nos activités d’enseignement, que ce soit au Master d’Informatique de l’Université Henri Poincaré – Nancy I ou dans celui des deux options de troisième année à Supélec, nous traitons des différents algorithmes de quantification vectorielle (k-means, cartes auto-organisatrices, réseaux incrémentaux, etc.). Pour asseoir notre pédagogie, nous avons conçu une bibliothèque permettant d’afficher des distributions 3D et la répartition des prototypes générée par les algorithmes, au fur et à mesure de la convergence (cf. figure 9.2).

Cet outil pédagogique s’appuie bien entendu sur ce que nous avons présenté au paragraphe 9.2, et est disponible via le *package* *DemoMapping* proposé sur notre site.

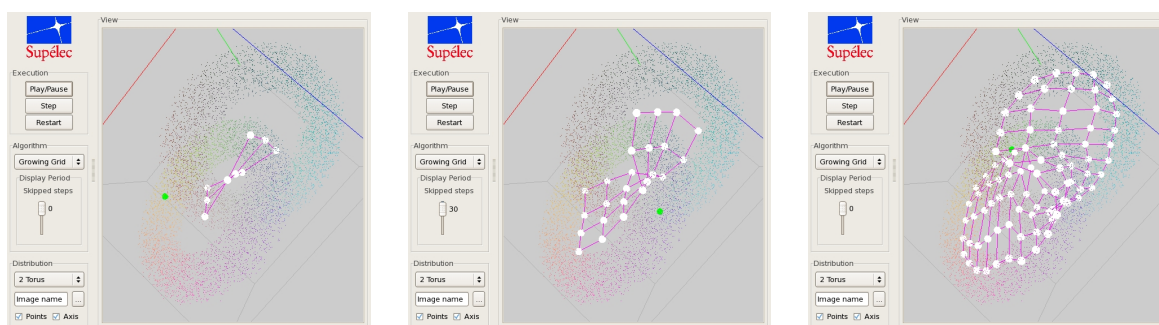


FIGURE 9.2 – Outil pédagogique pour l’enseignement de la quantification vectorielle.

9.5 Middleware réseau

Nous avons, dans notre laboratoire, acquis différents matériels pour notre plate-forme robotique et la *smartroom*. En général, les laboratoires qui réalisent des expérimentations robotiques éprouvent le besoin de concevoir une plate-forme logicielle permettant d’intégrer ces robots. Plutôt que de reprendre des logiciels existants, pas toujours simples à installer, nous avons choisi d’extraire de ces plate-formes ce qui en est le cœur, à savoir la mise en réseau de ressources. Nous proposons donc une architecture logicielle très légère, mais puissante du fait de l’utilisation de patrons de classe. Cette architecture est une sorte de mini-CORBA, qui s’articule en deux

bibliothèques, `nono` et `bkbd`.

nono

La bibliothèque `nono` est un *Middleware* réseau permettant d'écrire facilement des applications client-serveur. Motivé initialement par le besoin de gérer notre plate-forme robotique, `nono` est vite devenu la clé de voûte de tous nos développements réseau. Olivier Rochel et Nicolas Rougier ont pris part à la définition de `nono`, que nous avons réalisée ensuite.

`nono` s'appuie sur l'excellente bibliothèque `common c++` du projet GNU. `nono` permet de définir un serveur qui héberge des ressources, qui chacune sont des serveurs TCP/IP multi-threadés. Pour définir ces ressources, le concepteur n'a à sa charge que la conception de classes pour les requêtes et pour les réponses, les clients et serveurs s'écrivant en passant les types de requêtes et de réponses à des patrons de classes fournis par la bibliothèque. `nono` gère également les ports série, dont l'ouverture et la fermeture sur un ordinateur peuvent être pilotés par le réseau, ce qui est très pratique dans une application robotique.

Enfin, lors d'une collaboration avec Stéphane Vialle, Fabrice Sabatier, durant son stage de fin d'étude du CNAM, a ajouté à `nono` la possibilité de gérer des clients redondants. Ainsi, l'on peut piloter un robot par deux instances d'un même client qui le commande. Ces clients déroulent le même programme. Le premier qui effectue la commande provoque l'exécution du robot, et la réponse est enregistrée sur le serveur. Le deuxième client, en retard, qui en est à ce stade du programme, reçoit la réponse enregistrée, et « *croit* » que le robot lui a répondu. Ainsi, quand l'un des clients s'effondre, disons celui qui est en avance, l'autre laisse le serveur rejouer les étapes qu'il a « *en retard* », mais il reprend effectivement la main quand il a rattrapé son retard. Cette fonctionnalité, que `nono` peut fournir, a été utilisée dans notre équipe par notre collègue Stéphane Vialle, pour des études de la gestion multi-sites d'une plate-forme robotique [Vialle *et al.*, 2005].

bkbd

Au dessus de `nono`, nous avons souhaité définir la notion de tableau noir partagé, où diverses applications postent et lisent de l'information. Ainsi, Lucian Alecu, lors de son stage de fin d'étude à Supélec [Alecu, 2005], a écrit la bibliothèque `bkbd` (pour *black board*). Cette bibliothèque s'appuie elle aussi sur les patrons de classe C++, ce qui la rend très générale. Elle offre des fonctionnalités de double buffering, utiles lorsque l'on met à disposition sur le tableau noir une image provenant d'une caméra. Elle propose également des outils pour stocker et échanger des images, via le réseau, en mode compressé ou non. Les bibliothèques `axisPTZ` et `axisbkbd` mettent à disposition les images et les commandes des caméras sur le tableau noir. Une image peut être visualisée périodiquement par des clients, fournis par le *package* `bkbdboard`. Enfin, nous avons inclus dans le tableau noir l'interface `V4L` (*video for linux*) afin d'offrir à des clients l'image d'une caméra installée sur un ordinateur (*packages* `V4L-tiny-grab` pour acquérir une image vidéo sous linux, et `V4Lbkbd` pour mettre un flux vidéo à disposition de clients).

Ces bibliothèques permettent, en programmant simplement une fonction `main` en C++, de construire un serveur dédié à une plate-forme robotique. Nous proposons un utilitaire (*package* `supelecboard`) qui évite cette programmation, la plate-forme étant décrite dans un fichier XML, ce qui permet de construire dynamiquement le serveur qui correspond à la plate-forme. À titre d'exemple, nous donnons l'architecture de notre plate-forme de video-surveillance à Supélec (cf. figure 9.3). Cette plateforme est signalée aux personnes, elle est contournable. Elle ne fait l'objet d'aucun enregistrement, et le résultat des traitements est visualisé, de sorte que les per-

sonnes qui se soumettent à la plateforme voient le résultat des algorithmes, et puissent « jouer » avec.



FIGURE 9.3 – La bibliothèque `bkbd` pour la plate-forme de vidéo-surveillance du campus de Metz de Supélec. Le serveur héberge 4 buffers d'image, munis d'un système de double-buffering. Le premier buffer (gauche) est lu par un client qui en visualise le contenu sur l'écran. Les trois autres sont connectés aux caméras, dont ils récupèrent l'image. L'exécution d'un traitement se fait sur une autre machine, qui dispose d'un client pour lire les images depuis un des buffers et poster le résultat du traitement sur le buffer qui est visualisé. Ainsi, le concepteur de l'algorithme ne se soucie ni de visualisation, ni du déploiement de la plate-forme.

9.6 La bibliothèque `grumpf`

Nous avons proposé au paragraphe 4.4.3 la définition d'un calcul à grain fin, basé sur des unités qui mettent à jour des variables réelles, et qui exposent ces variables en lecture aux autres unités qui lui sont reliées. La mise en œuvre de cet environnement de calcul, que nous avons baptisé `grumpf` (Graphical Utilities for the Modelling of Parallel Functions), a été le développement qui a permis de faire progresser le plus notre projet de recherche. Cette bibliothèque se compose d'un noyau, qui réalise le calcul à grain fin, et qui soumet ce calcul à l'interrogation de clients qui peuvent lui envoyer de l'information ou observer son état. La bibliothèque permet au modélisateur de programmer un calcul ayant la propriété d'être un serveur. Les clients, eux, ne font pas l'objet d'un effort de programmation du modélisateur, ils lui sont fournis comme utilitaires par le `package grumpf`.

9.6.1 Le noyau `grumpf`

Le noyau `grumpf` est une bibliothèque C++ qui se fonde sur l'approche objet pour proposer à l'utilisateur la possibilité de programmer des unités, de les connecter et de lancer le calcul, à l'instar de ce que nous avons avancé au paragraphe 4.4.3. L'intérêt de l'approche objet réside dans la capacité pour l'utilisateur de définir son modèle sans se soucier de différents aspects bien

plus techniques. Le premier avantage est que son modèle est, sans effort de sa part, un serveur TCP/IP⁸¹ interrogeable par des clients graphiques (cf. figure 4.10), assurant la visualisation, mais aussi le déroulement pas à pas et la sauvegarde du réseau. Le second, et non le moindre, est la prise en charge d'une exécution parallèle du modèle par `grumpf`. À l'heure actuelle, cette exécution parallèle est effective sur machines multi-processeurs à mémoire partagée, via la bibliothèque `ParCel16` développée par Stéphane Vialle. Le portage du noyau sur cluster de PC est l'un des objectifs majeurs du projet `InterCell` présenté au paragraphe 8.1.2.

`grumpf` est fourni avec une série d'utilitaires clients, qui permettent la sauvegarde périodique d'une exécution, l'affichage en mode texte de l'activation du système, l'affichage en *ray-tracing* via `povray`, etc.

9.6.2 Les outils de visualisation

Les outils de visualisation que nous proposons ont fait l'objet d'un développement conséquent, pour lequel nous avons toutefois été accompagné. Tout d'abord, le système d'échange d'événements entre les clients et le serveur (notifications d'affichage, etc) a été conçu lors du stage ingénieur d'Anca Ghițescu [Ghițescu, 2004]. Deux élèves de Supélec, Florent Grenier et Louis Deflandre, ont également réalisé les « sondes », qui sont des objets graphiques pouvant visualiser les messages produit par une unité particulière. Enfin, la majeure partie des utilitaires graphiques de `grumpf` a été réalisée par Jacques Weidig, au cours de son stage d'ingénieur CNAM (12 mois) [Weidig, 2005].

9.6.3 `bijama`

La bibliothèque `bijama` s'appuie sur `grumpf` pour instancier le modèle que nous avons présenté au paragraphe 4.5. Nous en aborderons ici quelques points techniques. Le premier est que cette bibliothèque C++ utilise massivement les patrons de classe et la méta-programmation. Le concepteur peut ainsi en quelques lignes invoquer la construction par le pré-processeur C++ d'objets complexes, dont la cohérence est assurée par le fort typage des patrons de classes de C++. La faculté qu'ont les étages `bijama` de savoir s'écrire en `metapost` (cf. figure 4.11) est un garde-fous puissant lors de la mise au point de modèles qui, à force d'empilement d'étages, peuvent devenir complexes. C'est donc une gestion de cette complexité logicielle qu'offre la méta-programmation qui soustend l'interface que nous proposons au modélisateur avec `bijama`.

9.6.4 `lgn2v1`

Les bibliothèques `lgn2v1` et `lgn2grumpf` sont des bibliothèques de traitement d'image C++ qui mettent en œuvre les déformations de l'image induites par les projections des deux champs visuels sur la rétine (cf. figure 9.4). Nous avons utilisé cette bibliothèque pour des expérimentations d'auto-organisation de filtres sur des *images réelles*, lors du stage de Master de Jacques Henri [Henri, 2005]. Nous avons mesuré à cette occasion la difficulté d'une part de réaliser ces études sur des images réelles, contrairement à ce que font de nombreux auteurs, et d'autre part la nécessité de pouvoir simuler un grand nombre d'unités, d'où en partie la motivation pour le projet `InterCell` (cf. paragraphe 8.1.2).

81. Via la bibliothèque `nono`.

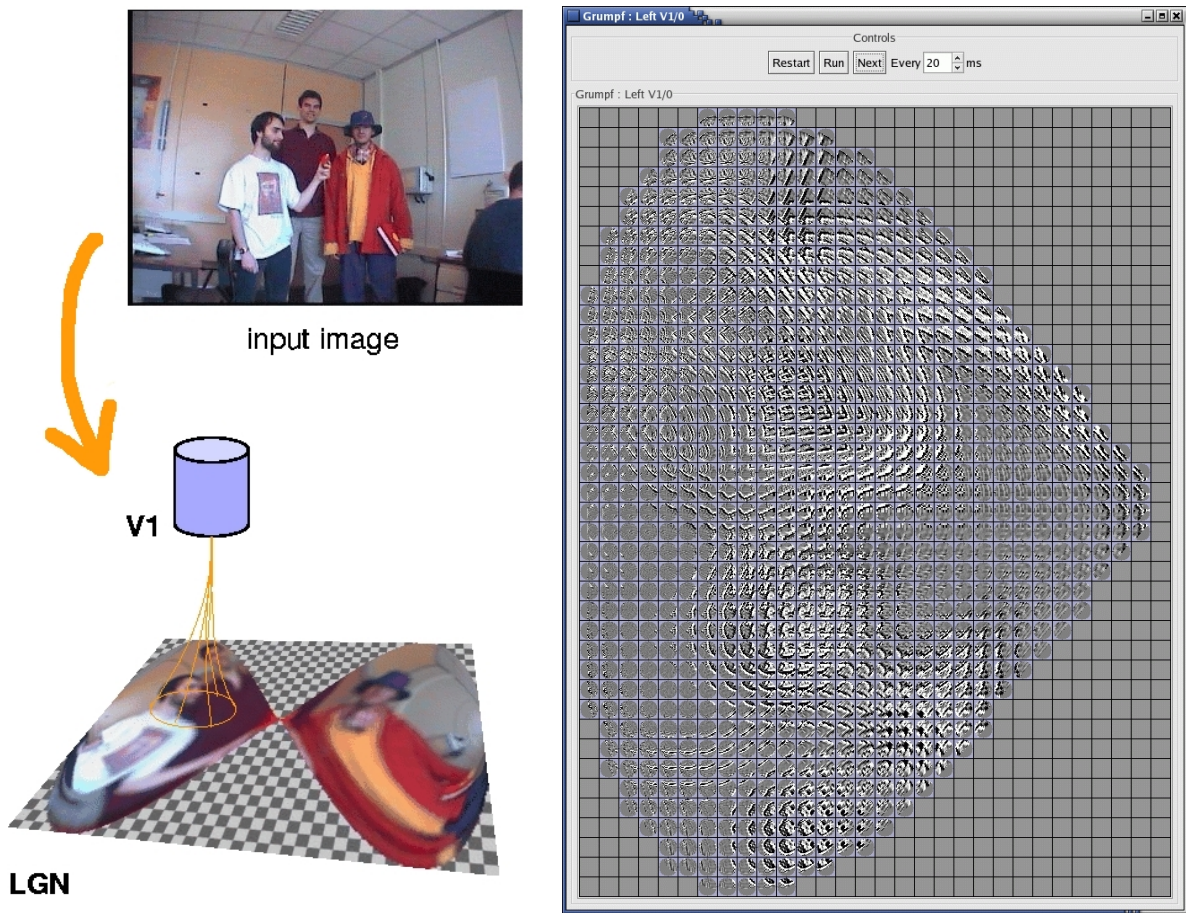


FIGURE 9.4 – Traitements réalisés par la bibliothèque `lgn2grumpf`. À droite, visualisation des champs récepteurs d'un module visuel primaire `grumpf` fourni par ces outils.

9.6.5 Le convertisseur simpf

Le logiciel `simpf` est un convertisseur qui, à partir de fichiers textes décrivant un automate cellulaire, génère le code `grumpf` qui réalise l'implémentation de cet automate. Les fichiers textes (fichiers `.simpf`) décrivent aussi bien l'état initial, les dimensions, que le type de mise à jour réalisé en chaque point de l'automate. Les fichiers `.simpf` n'ont pas vocation à être écrits par un humain, bien que cela soit faisable; ce sont les fichiers produits par `Mathematica` lorsque l'on utilise la suite logicielle `ESCAPaDE`, pour réaliser la simulation d'une équation aux dérivées partielles (cf. paragraphe 6.1). Les toutes premières versions de `simpf` ont été écrites par Jérémie Even et Loïc Salmona lors d'un projet de fin de deuxième année à Supélec.

9.7 La bibliothèque booz

9.7.1 Portage du calcul cellulaire interactif sur cluster

Notre contribution principale au projet `InterCell` (cf. paragraphe 8.1.2) est le développement de la bibliothèque `booz`, qui permet d'implémenter sur un cluster l'ensemble des concepts proposés par la bibliothèque `grumpf` présentée au paragraphe 9.6. La parallélisation du calcul cellulaire offerte par `booz` s'appuie sur l'usage de la bibliothèque `parXXL`, développée par Jens Gustedt et Stéphane Vialle [[parXXL, 2010](#)]. `booz` ajoute à ces fonctionnalités un système de référencement des unités ouvert et utilisable à grande échelle, ainsi que la possibilité d'interagir avec un calcul en cours, à des fins de visualisation mais aussi pour interfacer le calcul avec un dispositif robotique s'exécutant en ligne.

9.7.2 Le référencement paresseux de booz

Lorsqu'il s'agit de manipuler un grand nombre d'unités de calcul (plusieurs centaines de millions), il n'est plus possible de leur attribuer une référence (comme par exemple un numéro d'identifiant) que l'on stockerait dans une table. En effet, cette table de plusieurs millions d'identifiants occuperait une place mémoire non négligeable et vite rédhibitoire.

La solution que nous proposons et que nous avons implémentée dans la bibliothèque `booz` est d'utiliser des *références paresseuses*, qui sont calculées à la demande plutôt que stockées dans une table. Par exemple, si l'on décide d'identifier des unités par un entier, l'intervalle $[100, 1000[$ est une notation concise (les deux entiers 100 et 1000) pour représenter les 900 références 100, 101, \dots , 998, 999. L'énumération de ces 900 référence peut s'obtenir à la demande, à partir d'un calcul itératif simple, d'après les deux entiers 100 et 1000. La généralisation de ce principe d'évaluation paresseuse des références d'après une description est à la base de toutes les opérations fournies par `booz`, et le système de référencement (entiers, coordonnées 3D, etc...) est laissé à discrétion de l'utilisateur.

Soit \mathcal{R} l'ensemble dans lequel sont prises les références de chacune des unités de la simulation. Cet ensemble est déterminé par l'utilisateur. Notons $\text{Desc}(\mathcal{R})$ l'ensemble des *descriptions* élémentaires de « *paquets* » de références. La notion de description est également à définir par l'utilisateur, qui doit définir comment l'on peut à partir d'une description produire itérativement l'ensemble des références qu'elle contient. Un intervalle d'entiers peut jouer le rôle d'une description pour des références entières de notre exemple. D'après la donnée de l'ensemble des descriptions, `booz` offre la notion de *désignation*, notée ici $\mathcal{D}_{\text{Desc}(\mathcal{R})}$, qui est construite d'après les propriétés suivantes :

$$\begin{aligned}
 \emptyset &\in \mathcal{D}_{\text{Desc}(\mathcal{R})}, \text{ ALL} \in \mathcal{D}_{\text{Desc}(\mathcal{R})} \\
 \text{Desc}(\mathcal{R}) &\subset \mathcal{D}_{\text{Desc}(\mathcal{R})} \\
 (A, B) \in \mathcal{D}_{\text{Desc}(\mathcal{R})}^2 &\Rightarrow A \cup B \in \mathcal{D}_{\text{Desc}(\mathcal{R})} \\
 (A, B) \in \mathcal{D}_{\text{Desc}(\mathcal{R})}^2 &\Rightarrow A \cap B \in \mathcal{D}_{\text{Desc}(\mathcal{R})} \\
 \forall A \in \mathcal{D}_{\text{Desc}(\mathcal{R})}, A \cup \text{ALL} &= \text{ALL} \cup A = \text{ALL} \\
 \forall A \in \mathcal{D}_{\text{Desc}(\mathcal{R})}, A \cap \text{ALL} &= \text{ALL} \cap A = A
 \end{aligned} \tag{9.1}$$

La bibliothèque `booz` fournit un système d'évaluation paresseux qui permet par le calcul d'obtenir itérativement toutes les références contenues dans une désignation⁸². Ce système est utilisé en interne pour gérer les échanges d'information entre les neuds du cluster. Il est également explicitement utilisé par l'utilisateur pour définir la répartition des unités sur les différents noeuds du cluster. Nous renvoyons à la documentation de `booz` pour plus d'informations [InterCell, 2008].

9.7.3 Interactivité

Le système de référencement est utilisé massivement lors des interactions entre le calcul en cours sur le cluster et le monde extérieur. En effet, un protocole d'échange a été défini pour communiquer avec certaines cellules. Les cellules avec lesquelles l'on souhaite communiquer sont spécifiées par un argument lors des messages du protocole, cet argument étant la sérialisation d'une désignation.

L'interactivité, comme nous l'avons dit, permet une visualisation en cours *d'une partie* des cellules, à savoir celles qui sont désignées par le client de visualisation (cf. figure 9.5). Elle permet également de communiquer avec certaines cellules de la simulation lorsque le système simulé intervient dans une boucle robotique par exemple.

9.7.4 Exemples de références

Nous proposons deux exemples de références dans `booz`, étant entendu que l'utilisateur peut définir d'autres systèmes de références à sa guise.

Le premier système est l'hypercube de dimension n . Dans ce cas, l'ensemble des références est $\mathcal{R} = \mathbb{N}^n$. Nous avons choisi $\text{Desc}(\mathbb{N}^n) = \mathbb{N}^n \times \mathbb{N}^n$. Un ensemble élémentaire de références est alors défini par deux n -uplets d'entiers, que nous interprétons comme les coordonnées des sommets opposés d'un hypercube. Ce système est utilisé pour la résolution d'équations différentielles.

Le second système que nous proposons est un système où les unités sont regroupées dans des « cartes » bi-dimensionnelles. Dans ce cas, $\mathcal{R} = \text{string} \times \mathbb{N} \times \mathbb{N}$, ce qui signifie qu'une unité est repérée par le nom d'une carte et sa position 2D dans cette carte. Ce système est utilisé dans les simulations de réseaux de neurones d'inspiration corticale que nous développons.

9.8 Logiciels pour l'apprentissage automatique

Dans les enseignements que nous dispensons à Supélec ainsi qu'au Master Informatique de l'Université Henri Poincaré – Nancy I, nous traitons de l'apprentissage automatique numérique, supervisé et non supervisé. Nous avons profité de ces enseignements pour acquérir une compétence opérationnelle sur ces domaines, qui se réalise sous la forme de bibliothèques. Par rapport aux autres bibliothèques que nous avons trouvées, celles que nous proposons mettent l'accent sur

⁸². Seule la désignation ALL n'est pas itérable.

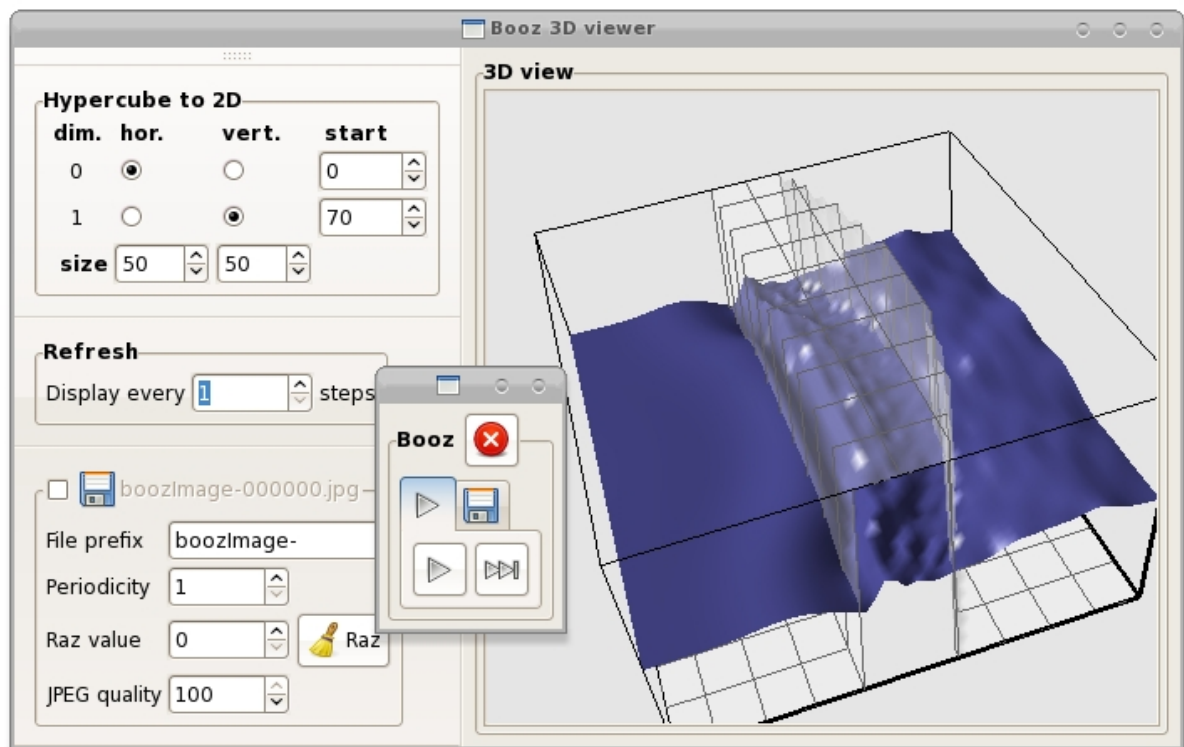


FIGURE 9.5 – Visualisation par un client graphique d'une partie de la simulation d'une onde traversant une lentille. La partie visualisé se limite à $[0, 50] \times [70, 70 + 50]$. La zone visualisée peut être changée dynamiquement depuis l'interface.

la qualité de l'interface (classes, paramétrisation par patrons de classe, etc.), qualité qui manque parfois aux produits de recherche mis à disposition de la communauté.

9.8.1 SVM et méthodes à noyaux

Nous proposons une suite logicielle pour la programmation de méthodes basées sur les noyaux, la bibliothèque `krnl`, sur laquelle nous avons fondé une bibliothèque `svm` fournissant deux machines à vecteurs supports, une pour la régression et une pour la classification. Les algorithmes d'optimisation sont ceux proposés dans [Keerthi *et al.*, 1999; Shevade *et al.*, 1999]. À l'heure où nous écrivons ces lignes, la conception de la bibliothèque est étendue au calcul de la plus petite sphère englobante, ce qui permet de réaliser un apprentissage non supervisé. Cette extension est l'objet du projet de fin d'étude de Sébastien (Euvrard, élève de troisième année).

La bibliothèque `krnl` fournit des outils pour réaliser des bases d'exemples, des procédures de validation croisée, etc. Cette bibliothèque fournit de plus des utilitaires pour réaliser un rendu `metapost` des résultats des algorithmes, comme l'illustre la figure 9.6. Ce module est très utile dans la réalisation de documents pédagogiques, par exemple.

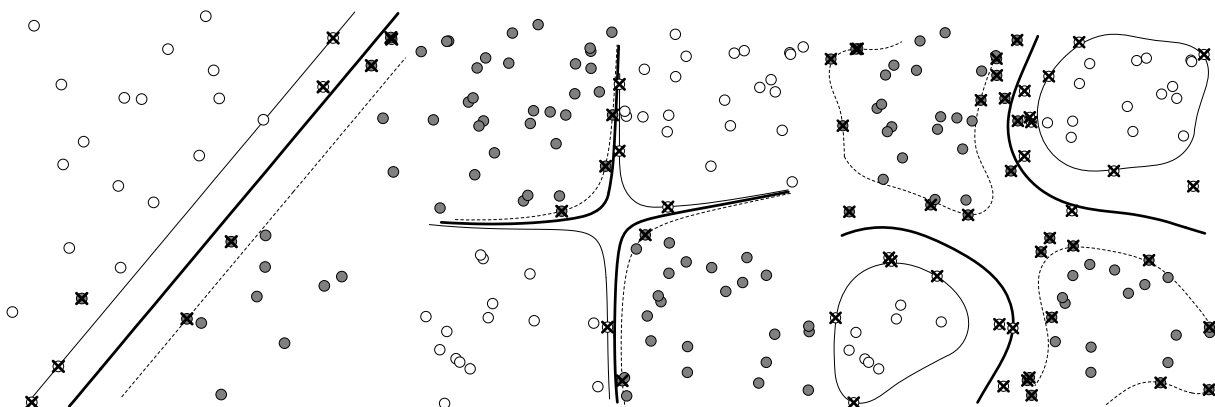


FIGURE 9.6 – Rendu `metapost` d'une classification par C-SVM, tel qu'il est fourni par la bibliothèque `krnl`.

9.8.2 GNG-T

L'algorithme de quantification vectorielle de distributions non stationnaires GNG-T que nous avons présenté au paragraphe 6.2 a lui aussi fait l'objet d'un développement logiciel pour le rendre paramétrable et opérationnel. Nous proposons pour ce faire une bibliothèque C++ nommée `vq`, qui rassemble les fonctionnalités communes à la plupart des algorithmes de quantification vectorielle, ainsi que la gestion de graphes très dynamiques (optimisation d'allocation mémoire, étiquetage stable des composantes connexes, etc.). C'est au-dessus de cette bibliothèque qu'est programmée la bibliothèque `gngtlib`, qui permet de mettre en œuvre très facilement cet algorithme. Nous renvoyons au paragraphe 6.2 pour plus de détail sur la nature de l'algorithme GNG-T.

9.9 Plate-formes robotiques et *smartroom*

À l'aide des *middleware* réseau présentés au paragraphe 9.5, nous avons interfacé les matériels que nous avons acquis de sorte qu'ils soient pilotables à distance par des interfaces C++ de haut

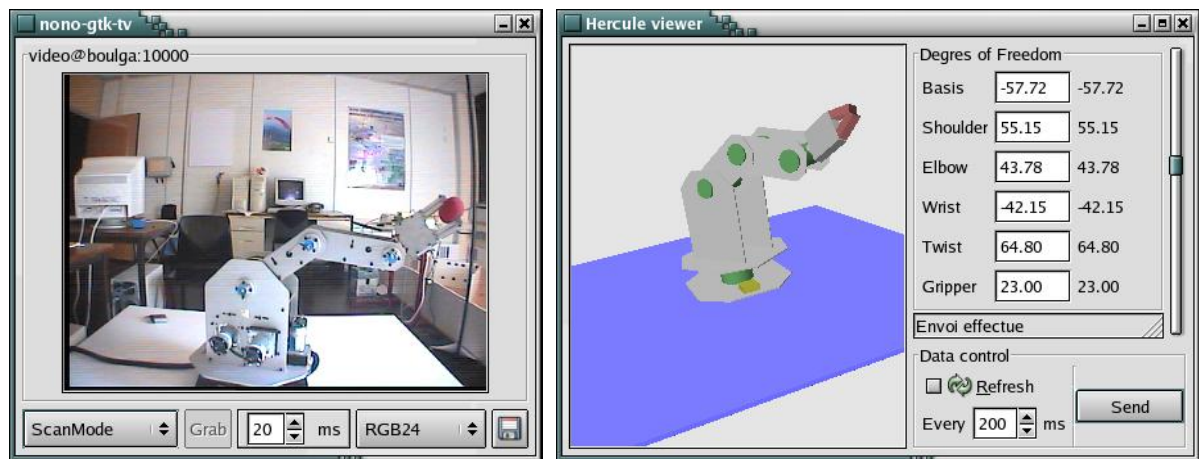


FIGURE 9.7 – Clients pour la gestion en réseau du robot *Hercule* de notre plate-forme. À droite, une interface *OpenGL* qui s’appuie sur la bibliothèque *glop* (cf. paragraphe 9.2) permet de contrôler le bras. Cette interface a été développée par des élèves de deuxième année de Supélec, Pierre Baranne et Julien Millot.

niveau. Nous avons réalisé cette interface pour les 2 robots *Koala* que nous possédons (avec leurs tourelles), pour les vidéos acquises par les cartes *bttv*, pour le bras articulé *Hercule*, et pour les 4 caméras *axis PTZ-213* dont nous disposons. Ces matériels sont visibles sur la figure 8.1, et la figure 9.7 montre une interface pour le bras articulé, cliente des serveurs *TCP/IP* que nos travaux permettent de déployer. Sont également inclus dans cette plate-forme une carte son multi-voies pour la localisation de sources sonores à partir de plusieurs microphones, ainsi qu’un système *EEG* mobile pour réaliser des interfaces cerveau-ordinateur.

Supélec, dans le cadre de l’intégration récente d’une partie des activités informatiques du campus de Metz au sein de l’UMI 2958 CNRS / Georgia Tech, met particulièrement l’accent sur le développement de la plate-forme *smartroom*. Outre le fait que nos recherches y sont naturellement incluses du fait qu’elles abordent la problématique de l’approche située, notre participation s’exprime également sous la forme d’une production logicielle conséquente, réalisée en collaboration avec Jean-Louis Gutzwiller et Jean-Baptiste Tavernier.

Troisième partie

Perspectives

Chapitre 10

Projet de recherche

Le projet de recherche que nous présentons dans ce document est l'affirmation de notre ambition de donner corps aux idées formulées au chapitre 1, en défendant la pertinence d'un support neuro-mimétique à l'intelligence artificielle. Ces idées, que nous souhaitons instancier dans nos systèmes, sont relatives aux approches situées et au caractère fondamentalement moteur de la cognition, dans ses composantes les plus abstraites tout comme dans celles qui sont les plus perceptives. Elles sont formulées par des biologistes et des philosophes, entre autres, mais peut-on vraiment être certain que leur élégance ne soit pas un leurre, qu'elles ne soient attractives que parce qu'elles s'opposent à l'approche symbolique de l'intelligence qui a dominé notre discipline, ou alors parce qu'elles bousculent une culture informatique traditionnelle qui, sur la question de calculer une intelligence, n'a pas su apporter de réponses satisfaisante ?

Tant que leurs partisans n'auront pas fait la preuve de la pertinence de cette démarche pour guider la conception d'une intelligence artificielle, ces idées restent des hypothèses de travail flottantes, en proie à toutes les contestations par des opinions philosophiques contraires (formuler ces idées a au moins cet intérêt-là). Leur donner corps, c'est pour nous et bien d'autres chercheurs les incarner dans l'exécution d'une machine de Turing, machine elle-même incarnée dans un système physique en interaction avec le monde, prête à passer un test, de Turing lui-aussi.

Avons-nous réussi cette incarnation par les modèles que nous proposons ? Que très partiellement, certes. Mais nous avons fait un petit pas dans une direction encore pleine d'ouvertures, ce qui laisse à notre discipline un espoir quant à cette question, elle qui s'y casse, depuis Turing, douloureusement les dents. Contrairement aux pionniers de l'intelligence artificielle qui s'appuyaient sur la psychologie, le « mouvement neuro-mimétique », avec l'aide des avancées en biologie, peut se soutenir de l'existence d'un système de traitement de l'information, le cerveau, qui, chez les êtres vivants, *calcule leur intelligence*.

Du fait de cette assurance, le petit pas que nous avons fait, apparaît comme capable d'en appeler d'autres, le long d'un chemin, certainement tortueux, nécessitant parfois des rebroussements, mais conduisant à la démonstration de la calculabilité de notre intelligence, de notre expérience de la conscience.

Dans ce chapitre, nous ne nous projeterons pas autant en avant sur ce chemin, préférant montrer quelles sont les orientations que nous souhaitons donner à nos recherches, qui motivent le choix des sujets que nous aborderons, de nos futures collaborations et des travaux de thèses que nous souhaitons accompagner.

10.1 Cheminement

10.1.1 Bilan rétrospectif des travaux de thèse (1999)

Nos travaux de thèse ont posé les bases de principes informatiques que nous suivons toujours aujourd'hui. Principalement, nous nous sommes imposé de concevoir des systèmes composés d'unités qui s'évaluent via des règles *strictement locales*, et dont les échanges d'informations sont des *lectures* de l'état d'autres unités. C'est ce principe de calcul, décrit au paragraphe 4.4.3, que nous avons mis en avant dans le projet *InterCell* présenté au paragraphe 8.1.2.

Nous avons également dans ces travaux contesté la notion de hiérarchie de modules corticaux, notion souvent présentée en biologie. Cette notion de hiérarchie est issue principalement d'interprétations du fonctionnement du traitement cortical de la vision : l'aire corticale *V1* extrairait les contours, l'aire *V2*, « *au-dessus* » de *V1* restituerait des lignes, jusqu'à ce que des neurones dits neurones « *grand-mère* », au sommet de cette hiérarchie, reconnaissent les formes. Nous avons développé au paragraphe 4.2.3 les arguments de notre contestation de cette la notion de hiérarchie au sujet de l'organisation du cortex, et le modèle que nous avons proposé durant notre thèse [Frezza-Buet and Alexandre, 2002] illustre que l'on peut répartir les différentes contraintes modales d'un comportement en différentes cartes, sans construire pour autant une architecture hiérarchisée. Cette contestation guide encore aujourd'hui nos travaux.

Nous avons également durant notre thèse fait la promotion d'une approche très motrice de la cognition, même pour ses aspects perceptifs puisque les modules visuels que nous avons proposés réalisaient une version réduite de la « *palpation par le regard* » proposée par Merleau-Ponty et les tenants de l'approche située (cf. paragraphe 1.3.2). Nous avons également, suivant cette vue motrice, jeté les bases d'une approche neuronale de la planification du comportement qui s'appuyait sur des mécanismes d'activités bistables, inspirés du cortex frontal, sur lesquels nous reviendrons dans ce chapitre. Ces travaux avait conduit à proposer des règles d'apprentissage de séquences robustes à la distorsion temporelle [Frezza-Buet and Alexandre, 1999; Frezza-Buet et al., 2001].

En revanche, dans les modèles que nous avons proposé alors, nous avons une approche que nous jugeons aujourd'hui trop localiste des activités de nos unités corticales⁸³. Cette vision localiste était le prolongement des travaux de Pascal Blanchet dont nous prenions la suite [Blanchet, 1994], travaux qui s'appuyaient sur le concept d'arbre d'appels proposés par Yves Burnod [Burnod, 1989] et modélisé par Frédéric Alexandre [Alexandre et al., 1991]. Dans notre modèle, la spécialisation d'unités initialement indifférenciées était implémentée par le clonage d'une unité « *germe* », produisant un clone dédié. Il s'avère que ce clonage complexifie énormément le modèle, d'un point de vue de la gestion informatique non centralisée que nous nous imposons. De plus, biologiquement, il modélise un processus de spécialisation qui relève de capacités d'auto-organisation et non de production de neurones.

En conclusion, ces travaux de thèses ont été l'occasion pour nous de poser des concepts qui guident encore aujourd'hui notre recherche de nouveaux paradigmes informatiques, et nous on laissé quelques frustrations qui ont fortement orienté la recherche que nous avons menée par la suite. Pour ces concepts autant que pour ces frustrations, nous souhaitons remercier ici Frédéric Alexandre de nous avoir permis de nous engager sur cette voie.

83. Cette remarque nous avait été faite en 1999 par Peter Ford Dominey lors d'une visite à son laboratoire, au moment où lui-même développait des réseaux de type réservoir.

10.1.2 Bilan des travaux à Supélec

Depuis que nous sommes à Supélec (mars 2000), nous avons pris le parti de mettre en suspend les études que nous avons amorcées sur le cortex frontal et la planification neuronale du comportement, car il nous a semblé que des problèmes relatifs à la maîtrise de la complexité des systèmes dynamiques devaient être abordés auparavant.

Nous avons donc abordé la question de l'auto-organisation de modules d'inspiration corticale régis par des champs de neurones, au cours des thèses d'Olivier Menard et Lucian Alecu [Mé-[nard, 2006](#); [Alecu et al., 2011](#)]. Ces travaux, que nous avons présentés au paragraphe 4.5, nous permettent aujourd'hui de nous affranchir du recours au clonage d'unités que nous utilisons durant notre thèse. Ils conduisent également à la représentation de l'information sous forme de population de neurones, ce qui n'est plus localiste et donc bien plus robuste.

D'un point de vue informatique, le fait de ne plus avoir à gérer le clonage de cellules a permis de simplifier la simulation de nos systèmes neuronaux. Nous avons, à Supélec, entrepris de collaborer avec des collègues chercheurs en parallélisme pour mettre à la disposition de nos recherches des outils de simulation. Initialement limité aux architectures à mémoire partagée (cf. paragraphe 9.6), notre simulateur est aujourd'hui disponible sur cluster de PC (cf. paragraphe 9.6.3) via le projet *InterCell*. Nous sommes aujourd'hui en mesure de simuler des systèmes dynamiques à grain fin comportant plusieurs millions d'unités.

Enfin, à Supélec, nous nous sommes efforcé de produire les logiciels nécessaires à la prise en main de matériels robotiques. Ces logiciels sont aujourd'hui utilisés et étendus dans le cadre de la smartoom et de la plate-forme robotique dont nous disposons sur le campus depuis mars 2010 (cf. paragraphe 9.9). Cet effort logiciel permet de mettre en œuvre la pédagogie des enseignements de Supélec auxquels nous participons, en particulier en troisième année dans l'option Systèmes Informatiques et Robotiques (cf. paragraphe 7.6.2), mais ils sont également impliqués dans la définition de plates-formes où nous illustrons nos recherches.

10.1.3 Travaux en cours

Actuellement, nos travaux de recherches portent sur l'auto-organisation de cartes corticales capables de prendre en charge des séquences. Ces travaux récents, qui font l'objet de la thèse en cours de Bassem Khouzam que nous co-dirigeons avec Frédéric Alexandre, visent à concilier la notion d'auto-organisation avec celle de réseaux récurrents de type réservoirs, apportant à ces derniers des propriétés d'adaptations. Il s'agit pour nous d'un retour aux problématiques d'apprentissage cortical temporel, problématiques que nous avons mises entre parenthèses depuis la fin de nos travaux de thèse, et que nous pouvons dorénavant reconsidérer à la lumière de nos récents résultats concernant les processus auto-organisants dirigés par des champs neuronaux.

D'autre part, nous menons une recherche plus appliquée en collaboration avec Jean-Louis Gutzwiller et Olivier Pietquin à Supélec, qui aborde la question de l'identification de locuteurs en ligne et en temps réel au fur et à mesure du déroulement du discours. Cette activité de recherche-là s'inscrit dans notre volonté de montrer que l'algorithme GNG-T que nous avons proposé, décrit au paragraphe 6.2, est un outil puissant d'apprentissage automatique.

10.1.4 Suite du projet de recherche

La suite du projet de recherche sur lequel nous souhaitons nous engager peut se décrire suivant différents points de vue. C'est ce qui est fait dans la suite de ce chapitre, où nous abordons le point de vue des fonctions cognitives que nous souhaitons réaliser, celui des architectures informatiques que nous souhaitons étudier, celui des systèmes dynamiques qui sont en tout état de cause ce que

nos recherches produisent, et enfin le point de vue des supports matériels permettant l'exécution des systèmes que nous nous proposons de concevoir. Nous ferons également mention de recherches « *collatérales* » qui font suite à ce que nous avons évoqué au chapitre 6, dans la mesure où ces recherches s'inscrivent naturellement dans le sillage de nos investigations concernant les systèmes non supervisés à grain fin.

10.2 Fonctions cognitives

Nous avons mentionné au paragraphe 1.2 que notre recherche porte sur la science informatique, et que c'est en poursuivant l'objectif de comprendre ce que peut ou non calculer une machine que nous pouvons apporter une contribution aux sciences cognitives, en complémentarité de ce qu'apportent biologistes, psychologues, philosophes et autres acteurs. Les fonctions cognitives, ou l'intelligence artificielle (voir la discussion du paragraphe 1.1), jouent pour nos algorithmes le rôle d'un contexte applicatif, alors même que les sciences cognitives sont aujourd'hui une problématique qui appartient à une recherche plutôt fondamentale.

Pour un informaticien, qu'il soit ingénieur ou chercheur, le contexte applicatif, quand il est retors comme l'est le contexte des sciences cognitives, est structurant. Autrement dit, pour nos recherches, poser un bon problème à résoudre, même si par certains de ces aspects ce n'est qu'un problème jouet, permet de faire avancer la conception de paradigmes de calculs nouveaux. Toutefois, et c'est là que le contexte applicatif n'en est pas un au sens des sciences de l'ingénieur, *ce n'est pas la résolution du problème qui nous importe, mais la façon de le résoudre.*

Nous nous proposons dans cette section de présenter les « *applications* » que nous souhaitons aborder dans le futur pour structurer la suite de nos recherches.

10.2.1 Champ d'application

Approche située

La notion d'approche située, que nous avons définie au paragraphe 1.3.1, est une application naturelle de nos recherches, du fait de l'inspiration biologique qui dirige la conception de nos systèmes informatiques. En effet, le cerveau humain est incapable de proposer instantanément un classement de plusieurs milliers de page web en fonction de la pertinence de quelques mots-clé, et il semble naturel de ne pas nécessairement recourir à des paradigmes informatiques comme ceux que nous proposons pour résoudre ces tâches. En revanche, dès qu'il s'agit d'inscrire un système de type robotique dans un environnement, l'approche inspirée du système nerveux est naturellement pertinente⁸⁴.

L'approche située est la direction dans laquelle s'est engagée notre équipe, l'équipe Information Multi-modalité et Signal, au sein de l'UMI 2958 CNRS/Georgia Tech à laquelle elle appartient depuis 2010. Cette approche se décline au sein d'une plate-forme nommée *smartroom* (cf. paragraphe 9.9) qui a fait l'objet d'une extension des bâtiments campus de Metz de Supélec. Nous souhaitons contribuer, par nos recherches, au rayonnement de cette plate-forme.

Construction de contrôleurs, apprentissage par renforcement

Le paradigme d'apprentissage par renforcement est central dans nos recherches, comme nous l'avons mentionné au paragraphe 5.3. Néanmoins, notre recherche ne porte pas sur les mathématiques liées à la formalisation de ce problème, formalisation issue des équations de Bellman et

84. Ceci n'exclut nullement d'en proposer d'autres, bien entendu.

de concept de différences temporelles. Toutefois, le domaine de l'apprentissage par renforcement et plus généralement du contrôle nous concerne, d'un point de vue applicatif, et l'orientation de nos recherches vers la construction de comportements nous amène vers ces paradigmes⁸⁵. Pour nos recherches, ils servent de référence, car il s'agit d'une formalisation remarquable des problèmes de contrôle qui nous intéressent. Nous entretenons, au sein de notre équipe, des échanges fréquents avec Olivier Pietquin et Mathieu Geist sur ces questions, ainsi qu'avec Alain Dutech au Loria à Nancy.

Toutefois, le paradigme d'apprentissage par renforcement est contestable à plusieurs titres, et c'est là tout l'intérêt que l'on peut tirer de sa formalisation mathématique. Tout d'abord, il suppose un critère de récompense réduit à une fonction scalaire, qui joue un rôle central dans la définition de ce que calculent les algorithmes d'apprentissage par renforcement⁸⁶. Rien ne permet aujourd'hui à notre connaissance d'affirmer qu'un animal aux motivations multi-valuées (faim, soif, sex, nouveauté, etc.) intègre ces motivations en interne sous forme d'un unique scalaire. Le deuxième point de contestation est celui des échelles de temps. Nous détaillerons plus avant notre volonté d'approfondir la question des échelles de temps dans l'apprentissage du comportement, en soulignant la nécessité de faire collaborer des mécanismes d'apprentissage lent (cf. paragraphe 10.4.1) et des mécanismes d'apprentissage immédiat (cf. paragraphe 10.4.2). Cette question n'est pas prise en compte à notre connaissance dans les recherches actuelles sur l'apprentissage par renforcement, et nous souhaitons profiter de l'éclairage des chercheurs de ce domaine que nous côtoyons pour poser correctement ce problème et critiquer notre approche.

10.2.2 Propriétés à explorer

Conjointement aux « *applications* » futures de nos travaux que nous venons d'évoquer, nous souhaitons aborder dans les années qui viennent certaines propriétés nécessaires à la cognition artificielle. Donner une liste exhaustive des propriétés attendue de la cognition reviendrait à en fournir une spécification, ce que nous avons soutenu comme étant inconsistant au paragraphe 1.1. Nous nous contenterons donc dans ce paragraphe de n'aborder que les deux points sur lesquels nous pensons pouvoir avancer dans un avenir proche.

Prise en compte du *temps physique* et du *temps logique*

Comme nous l'avons présenté au paragraphe 5.2, nous pensons qu'il est pertinent de différencier le temps physique du temps logique. Le temps physique est celui, compté en secondes, selon lequel se déroulent les processus mécaniques ou inertiels. Il intervient, par exemple, dans le contrôle d'un pendule inversé, problème souvent utilisé en apprentissage par renforcement. Le temps logique, lui, est le temps suivant lequel s'articulent les différentes étapes d'un raisonnement. Par exemple, apprendre à un système artificiel à jouer au backgammon est aussi un problème étudié en apprentissage par renforcement, mais les instants précis auxquels jouent les joueurs n'ont pas d'importance.

De façon générale, le temps logique est celui du déroulement d'un plan, alors que le temps physique intervient dans la gestion de l'équilibre, de l'inertie, au cours du déroulement de ce plan. Nous avons dans nos travaux de thèse abordé ces deux aspects, avec d'une part des règles d'apprentissage temporel liées au temps physique et d'autre part la gestion d'unités bistables, inspirées du cortex frontal, qui prenait en charge le temps logique. Nous souhaitons aujourd'hui

85. Nous enseignons d'ailleurs l'apprentissage par renforcement à l'Université, cf. paragraphe 7.6.2.

86. La notion de politique optimale est directement tributaire de la fonction de récompense que l'on définit a priori.

revenir sur ces notions en les séparant dans différents systèmes (inspiration du cervelet et du cortex moteur pour le temps physique, inspiration du cortex frontal et des ganglions de la base pour le temps logique) et en abordant la question de la coordination de ces différents systèmes au cours de l'apprentissage et de la réalisation d'un comportement.

Transferts d'apprentissage

Parmi les propriétés qui à notre sens sont un défi à relever par nos approches neuro-mimétiques, il y a eu la notion d'auto-organisation, abordée par Kohonen dans les années 80, et que nous avons pour notre part étudiée dans le contexte des champs neuronaux dynamiques lors de la thèse de Lucian Alecu. Il est une autre propriété qui nous semble connexe à l'auto-organisation, et qui constitue un défi pour l'informatique : le transfert d'apprentissage, que nous souhaitons aborder dans nos travaux futurs.

Lorsqu'il apprend une compétence motrice, comme jouer au tennis par exemple, le débutant réfléchit à chacun de ses mouvements, ce qui implique cortex frontal et ganglions de la base et relève d'une planification de gestes. Au fur et à mesure que le geste, à force de répétition, s'automatise, il s'engramme dans le cortex moteur, puis dans le cervelet où il devient un automatisme auquel on ne réfléchit plus. Le joueur expert parviendra même à engrammer ses gestes au niveau de la moelle épinière, le geste étant alors déclenché bien plus rapidement.

Ce qui est un défi à l'ingénierie classique dans cette histoire⁸⁷, c'est qu'au fur et à mesure que la gestion du mouvement passe des centres cognitifs vers les centres réflexes, les structures cognitives redeviennent disponibles pour l'apprentissage de nouveaux gestes. Comme pour l'auto-organisation, le recrutement de ressources cognitives (logiques) puis leur désengagement au profit de ressources plus automatiques (réflexes) s'effectue de façon non supervisée. Nous souhaitons creuser cette question plus avant, avec l'aide de la littérature biologique, de notre plate-forme robotique et de nos moyens de calcul (cluster `InterCell`).

10.3 Architectures informatiques

Les algorithmes que nous proposons pour aborder « *la cognition* » mettent en œuvre des populations d'unités de calcul, jouant ainsi sur les propriétés globales que l'on peut attendre de la somme des interactions d'unités au sein d'une population. Contrairement à ce qui se fait dans le domaine du *reservoir computing* que nous avons abordé au paragraphe 3.5, nous insistons dans nos travaux sur la structure des connexions entre unités. En effet, la biologie nous montre que le cerveau est, d'un point de vue anatomique, très organisé, très régulier. Nous mettons donc particulièrement l'accent sur la structures des connexions dans les réseaux que nous proposons, ce qui nous amène à concevoir des populations de neurones comme des *architectures neuronales* et non des réservoirs amorphes⁸⁸. Les architectures que nous souhaitons aborder dans la suite de nos recherches sont présentées dans ce chapitre.

10.3.1 Architectures extensives

Nous avons présenté en début de ce document, au paragraphe 2.2.2, la notion de ressource de calcul extensives. Il s'agit de définir les calculs « *au m²* » , laissant à la surface de calcul le soin de s'auto-organiser pour prendre en charge la tâche au sein de laquelle elle est impliquée. L'ensemble

87. dont nous demandons aux biologistes de bien vouloir nous pardonner les raccourcis.

88. Le caractère amorphe des réservoirs est ce qui fait leur intérêt, nous disons simplement ici que nous nous engageons sur une autre voie.

de ce document constitue à nos yeux un plaidoyer en faveur des architectures extensives, même si nous sommes conscients que nous n'avons par aujourd'hui fait la preuve que ces architectures ne sont pas qu'un vœu pieu. Nous souhaitons persévérer dans cette direction, avec nos collègues du Loria, en particulier Bernard Girau, Yann Boniface et Nicolas Rougier.

10.3.2 Inspiration d'autres grandes structures cérébrales 2D

Le cortex cérébral, dont nos travaux se sont inspirés jusqu'ici, est une structure laminaire bidimensionnelle (cf. paragraphe 4.1.1). Ce constat anatomique a été très structurant pour l'informatique, puisque la notion de champs neuronaux dynamiques en découle (cf. paragraphe 3.6.2), tout comme le concept connexe de cartes auto-organisatrices [Kohonen, 1997]. Nous envisageons bien entendu de continuer à nous inspirer du cortex cérébral, le sujet étant loin d'être épuisé, mais nous nous proposons également de considérer d'autres structures bidimensionnelles. En effet, d'autres centres nerveux tout aussi génériques et reconfigurables que le cortex existent dans nos cerveaux, comme par exemple l'hippocampe [Rougier, 2000], le cervelet, le striatum. Ces centres semblent toutefois relever d'une autre logique que celle de l'auto-organisation corticale, alors même qu'ils organisent des calculs sur une surface.

Quelle est la pertinence d'une organisation bidimensionnelle d'unités à grain fin ? Que peut-on faire avec ces structures ? Il y a-t-il une caractéristique fonctionnelle commune aux différents traitements qu'elles prennent en charge ? Sont elles une solution générale aux problèmes de latence (cf. paragraphe 2.1.3) ? Toutes ces questions sont directement dans le collimateur de nos travaux.

10.3.3 Structures 3D

Nous évoquerons également la possibilité d'envisager des structures neuronales tridimensionnelles, ce qui est permis par la généralité du système de référencement d'InterCell (cf. paragraphe 9.7.2). Cela concerne la modélisation de noyaux neuronaux, mais aussi la connexion de structures surfaciques entre elles, car cette connexion en biologie se doit de tenir dans la boîte crânienne. La considération de l'étalement spatial des calculs rejoint encore une fois la problématique de latence que nous avons déjà mentionnée (cf. paragraphe 2.1.3).

10.4 Systèmes dynamiques

Les systèmes dynamiques sont, de fait, ce que nous produisons au cours de nos recherches, et nous leur avons consacré le chapitre 3 de ce document. Nous présentons ici ce que nous pensons pouvoir proposer à l'avenir en terme de propriétés de systèmes dynamiques. Il s'agit pour nous, comme nous l'avons évoqué au paragraphe 10.2.1, de faire cohabiter des mécanismes d'apprentissage lent avec des mécanismes d'apprentissage immédiat.

10.4.1 Apprentissage lent

Nombre de paradigmes d'apprentissage reposent sur l'adaptation lente de poids, qui au fur et à mesure du déroulement d'un flux « *s'adaptent* » pour emmagasiner une information. Il s'agit en général d'extraire du monde une régularité immuable, une « *loi* », qui est encodée durablement par les poids.

Auto-organisation

L'extraction de lois est ce que nos modules d'inspiration corticale auto-organisants réalisent (cf. paragraphe 4.5), et nous souhaitons poursuivre dans cette direction. En particulier, le principe d'auto-organisation conjointe est à développer. En effet, ce principe a été établi lors de la thèse d'Olivier Ménard, puis la thèse de Lucian Alecu a proposé une équation de champs neuronaux assurant l'auto-organisation, alors qu'Olivier Ménard utilisait pour ce faire une équation difficile à paramétrer, que nous souhaitons reconsidérer. Aujourd'hui, il convient de concilier les résultats de ces deux thèses, afin de confronter l'équation proposée par Lucian Alecu au contexte d'auto-organisation conjointe. On pourra ainsi poursuivre l'approche architecturale proposée par Olivier Ménard, qui ouvrait la voie à la conception d'une architecture neuronale acteur-critique auto-organisée (cf. paragraphe 5.4.2). De plus, d'un point de vue dynamique, nous souhaitons étudier la notion d'auto-organisation conjointe sur une grande quantité de modules, selon différents connexions inter-modules. Cette étude est réalisable du fait de la puissance de calcul offerte par `InterCell`.

Réservoirs

Les réservoirs, présentés au paragraphe 3.5, sont également un moyen de représenter une loi du monde, cette loi étant cette fois-ci temporelle. Nous souhaitons pouvoir adjoindre au concept de réseau récurrent à délai temporel que sont les réservoirs la notion d'auto-organisation, de sorte à organiser la récurrence temporelle pour qu'elle « *assimile* » une régularité temporelle qui lui est soumise. Ce défi est ce qui est abordé actuellement par la thèse de Bassem Khouzam. L'enjeu est d'utiliser ces *cartes corticales temporelles* comme un outil qui puisse engammer une représentation markovienne à partir d'un flux d'événements qui, pris isolément, ne sont pas markoviens. Par exemple, la photographie d'une balle n'est pas markovienne, car il manque les informations relatives à son mouvement pour déterminer son état. Toutefois, soumise à une séquence de photographies successive de balles, nous attendons de nos cartes temporelles qu'elles puissent à chaque instant fournir un état qui, ayant intégré la séquence passée au sein de représentations temporelles récurrentes, soit caractéristique de l'état physique de la balle, vitesse incluse (état markovien).

10.4.2 Apprentissage immédiat

Parallèlement à l'apprentissage lent abordé au paragraphe précédent, d'autres mécanismes sont à considérer pour mettre en œuvre un apprentissage immédiat. L'apprentissage est un changement d'état du système, le changement concernant les poids en cas d'apprentissage lent. Pour un apprentissage immédiat, l'état qui est impacté est plutôt une *distribution d'activité*. En toute rigueur, qualifier d'apprentissage tout changement d'état d'un système peut paraître abusif, même si c'est bien de cela qu'il s'agit lorsqu'un système apprend. Nous dirons plutôt d'un changement d'état qu'il est un apprentissage quand le nouvel état a pour fonction de *moduler* ou de *biais* le fonctionnement du système. Ainsi, construire des activités modulatrices est un moyen de mettre en place instantanément un apprentissage, alors que modifier des poids comme précédemment est un moyen progressif et durable de le faire.

L'état d'activation immédiatement appris peut être vu comme la prise en compte d'un contexte, qui module la tâche courante. Dans le modèle pionnier de cortex frontal décrit dans [Dehaene and Changeux, 1989], ce contexte était même une activation aléatoire. Nous souhaitons pour notre part poursuivre nos recherches sur cette question en nous appuyant sur deux outils dynamiques, les unités bistables et la mémoire épisodique.

Bistabilité

Les unités bistables font parti des outils que nous avons étudié lors de nos travaux de thèse, et dont nous avons suspendu l'étude depuis afin de mieux comprendre la dynamique d'auto-organisation multi-modale. Il s'agit d'unités ayant deux états, actif et inactif, dont les transitions de l'un à l'autre sont consécutives à des événements. L'événement « *on* » fait passer l'unité d'inactive à active, et cette activité reste *soutenue d'elle même* jusqu'à nouvel ordre. Ce nouvel ordre est la survenue d'un événement « *off* », qui ramène l'unité à son état inactif.

Nous avons montré dans nos travaux de thèse que ces unités sont de bons candidats à l'implémentation d'une *pile neuronale* pour gérer l'articulation de buts et de sous-buts dans une opération de planification, comme rappelé au paragraphe 5.2.3, et il convient aujourd'hui de revenir sur ces questions à la lumière de la connaissance des systèmes dynamiques d'inspiration corticale que nous avons accumulée depuis ces travaux de thèse. Ces mécanismes relèvent de la gestion du *temps logique* nécessaire au systèmes cognitifs.

Mémoires épisodiques

Outre la notion d'établissement puis de maintien des buts courants du plan comportemental vue au paragraphe précédent, l'établissement immédiat de représentations visant à influencer le comportement courant concerne également la notion de mémoire épisodique. Nous n'avons pas traité cette question jusqu'alors, qui relève de la notion de mémoire auto-associative présentée au paragraphe 3.6.1, et relative à l'hippocampe, comme proposé dans la thèse de Nicolas Rougier [Rougier, 2000]. Il est toutefois clair qu'un système dynamique permettant de construire un comportement situé doit posséder une capacité de *mémoire épisodique*, c'est-à-dire la capacité à assimiler instantanément *un fait* important, imprévu, puis d'en tenir compte dans le déroulement futur du comportement. Il nous faudra un jour où l'autre aborder cette question, par exemple au cours d'une collaboration impliquant des modèles d'hippocampe.

10.4.3 Déséquilibre permanent ?

La notion de système dynamique abordée dans cette partie est concomitante à celle de flux de données. Le système dynamique, dans une approche située du contrôle, est en effet couplé à son environnement en permanence, au cours du temps. Ce couplage s'effectue au niveau du flux des entrées reçues par le système, mais aussi du flux des sorties, principalement motrices, qu'il produit. Calculer au sein d'un flux pose la question de la stabilité du système dynamique. Par exemple, on peut considérer à chaque instant que le système est en un point stable, et qu'il change au cours du temps de point de stabilité. Cela suppose qu'atteindre un état stable se fait à des constantes de temps bien plus rapides que celles suivant lesquelles se déroule les flux d'entrées et sorties. On peut également considérer que le système dynamique est en déséquilibre permanent. Suivant cette conception, il est en permanence en train de relaxer vers un point de stabilité, mais ce point change continuellement sans n'être jamais vraiment atteint.

Au regard de ce qui précède, une troisième conception peut être proposée, en complément des deux précédemment évoquées. Il s'agit de considérer que deux dynamiques coopèrent. Pour l'une comme pour l'autre, les échelles de temps peuvent être très variables, mais ce qui les différencie est le rôle que joue le temps, nous ramenant à la notion de temps logique et de temps physique.

Cela nous ramène à considérer au sein de la dynamique trois notions de temps qui se recouvrent : le temps de relaxation vers un état stable, le temps physique nécessaire au contrôle des phénomènes inertiels, et le temps logique au cours duquel s'articulent les raisonnements. Or pour une machine, comme pour n'importe quel être vivant d'ailleurs, seul le temps physique

compte, les deux autres n'étant qu'une interprétation de la part du concepteur de ce qui se joue dans la dynamique du système. Articuler ces différentes temporalités au sein d'une machine de Turing fait partie des problèmes épineux que nous avons à résoudre.

10.5 Supports de calcul

Notre recherche, au fur et à mesure de ses développements, requiert de pouvoir simuler des systèmes dynamiques de plus en plus complexes, impliquant de nombreuses unités et connexions. Cela nous impose de considérer la question du support physique sur lequel peuvent s'exécuter ces calculs.

10.5.1 Modèles non-impulsionnels

Nos travaux de thèse font résolument la promotion de systèmes dynamiques basés sur la mise à jour de grandeurs scalaires. Ils s'opposent en cela à des modèles impulsionnels, dont la simulation revient à gérer des événements (émission de *spikes*). Nos modèles sont dits fréquentiels, dans la mesure où le pendant biologique des scalaires que nous manipulons est la fréquence instantanée des trains de *spikes* qui sont émis par les neurones. L'usage d'une approche fréquentielle permet de contraindre, et donc de spécifier, la conception de supports de calcul parallèles permettant à nos modèles de s'exécuter. Ce type de contrainte existe également pour l'approche impulsionnelle, pour laquelle de nombreux simulateurs existent déjà (cf. paragraphe 3.8), alors qu'à notre connaissance, seul **InterCell** permet de simuler de grands systèmes dynamiques fréquentiels au sens où nous l'entendons.

10.5.2 Expansion physique des calculs

Concevoir à l'avenir de très gros systèmes à grain fin impose de prendre en compte les phénomènes de latence, comme décrit au paragraphe 2.1.3. En effet, pour ces gros systèmes, le temps de propagation de la lumière au sein des circuits est lent, et devient une limitation à la course aux performances dans laquelle la technologie s'est lancée. Nous avons vu que le cerveau est un exemple de structure qui calcule la cognition, ce qui légitime le fait de s'en inspirer pour réaliser une intelligence artificielle. Suivant ce même raisonnement, le cerveau est aussi une structure dont on peut s'inspirer pour résoudre les problèmes de latence, car il s'agit d'un système complexe, distribué, basé sur une technologie électrique très lente⁸⁹, pour lequel la répartition des calculs dans l'espace de la boîte crânienne est cruciale. Les solutions trouvées par l'évolution à cette problématique d'architecture des systèmes informatiques ont certainement quelque chose à apprendre aux architectes d'aujourd'hui, qui travaillent en général sur des composants bidimensionnels.

10.5.3 Suite de projet InterCell

La suite que nous allons donner au projet **InterCell** d'un point de vue logiciel est limitée⁹⁰ car notre participation se situe au niveau d'une bibliothèque, présentée au paragraphe 9.7, qui est aujourd'hui opérationnelle. Nous devrions pouvoir profiter des développements réalisés par nos collègues chercheurs en parallélisme pour continuer à porter nos architectures sur les machines futures.

89. La vitesse de propagation des signaux est d'environ $2\text{m}\cdot\text{s}^{-1}$.

90. Au delà des travaux de maintenance logicielle et d'amélioration.

Du point de vue de la modélisation neuronale, nous nous proposons d'étendre les outils offerts par nos bibliothèques, en terme de visualisation, mais surtout en termes de systèmes de référencement. En effet, positionner des unités au sein d'une grille, comme nous le faisons actuellement, pose des problèmes d'anisotropie dans certains modèles, problèmes que l'on peut résoudre par des références adéquates à développer.

10.5.4 Composants matériels génériques

On pourrait reprocher à nos travaux de ne pouvoir s'implémenter que sur des *clusters* de machine, alors qu'il existe d'autres paradigmes de machines parallèles. En particulier, des composants comme les FPGA et les GPU constituent des solutions d'avenir à la parallélisation.

Pour ce qui est des GPU, notre collègue Stéphane Vialle travaille sur ces machines, et il existe sur le campus de Metz de Supélec un *cluster* de GPU. Nous attendons que ces recherches en parallélisme soient suffisamment abouties pour porter nos travaux sur ces architectures, sachant que c'est ce qu'il s'est passé pour le portage sur *cluster* de PC.

Pour ce qui est des FPGA, nous collaborons avec Bernard Girau à la définition de ressources de calculs extensives (cf. paragraphe 2.2.2). Nous apportons à cette collaboration une formalisation de ce que sont les calculs de type cortical, que Bernard Girau cherche à exploiter pour proposer des composants qui réalisent, sur base FPGA, ces calculs génériques extensifs. Là aussi, les problèmes de localité des calculs font que l'approche que nous poursuivons est a priori compatible avec les contraintes physiques de routage sur les FPGA.

Pour résumer, nous ne souhaitons pas par nous-même nous impliquer dans le portage sur tel ou tel matériel de nos architectures, mais nous souhaitons entretenir nos collaborations avec des chercheurs concernés par ces problématiques.

10.6 Recherche collatérale

Terminons la présentation de notre projet de recherche par la suite que nous souhaitons pouvoir donner aux recherches « *collatérales* » que nous avons présentées au chapitre 6.

10.6.1 Résolutions d'équations différentielles

Le système de résolution d'équations différentielles sur le cluster *InterCell* que nous avons réalisé avec Nicolas Fressengeas est aujourd'hui opérationnel. Toutefois, il ne peut pas, par sa structure, prendre en charge des équations différentielles de type Vlasov, équations impliquées dans la modélisation de plasma. Suite à des discussions avec Alain Ghizzo, du laboratoire LPMIA à Nancy, nous pensons néanmoins pouvoir proposer via *InterCell* une procédure de résolution « *directe* » de ces équations. Cela implique une extension conséquente de la suite logicielle dont nous disposons actuellement, qui suppose de trouver un cadre au sein duquel nous puissions recruter un ingénieur dédié à cette tâche.

10.6.2 Applications de GNG-T

L'algorithme GNG-T que nous avons proposé au paragraphe 6.2 est un algorithme d'apprentissage automatique général, dont nous souhaitons montrer les capacités en l'appliquant à différents domaines. Le cadre de Supélec, en tant qu'école d'ingénieur, est propice à cette entreprise, car nous sommes amenés à participer à des projets industriels très applicatifs, au cours desquels ce type d'algorithme peut être utilisé.

10.7 Synthèse

La définition de notre projet de recherche, telle qu'elle se présente dans ce chapitre, revient à articuler la même problématique, celle du calcul cellulaire situé, en prenant des points de vue différents : point de vue fonctionnel, architectural, dynamique et matériel. Ces points de vue sont autant d'opportunités d'impliquer ce projet dans des collaborations pluridisciplinaires, desquels notre problématique s'enrichira, comme elle s'est enrichie des collaborations passées. C'est dans cet esprit que nous souhaitons proposer à la communauté scientifique un projet résolu, soutenant des hypothèses risquées mais stimulantes, qui a vocation à s'associer de façon complémentaire et la plus constructive aux travaux de chercheurs des disciplines que nous avons mentionnées dans l'ensemble de ce document.

Bibliographie

- [Abelson *et al.*, 1984] Hal Abelson, Jerry Sussman, and Julie Sussman. *Structure and Interpretation of Computer Programs*. MIT Press, 1984.
- [Abelson *et al.*, 2000] Harold Abelson, Don Allen, Daniel Coore, Chris Hanson, George Homsy, Jr. Thomas F. Knight, Radhika Nagpal, Erik Rauch, Gerald Jay Sussman, and Ron Weiss. Amorphous computing. *Communications of the ACM*, 43(5) :74–82, 2000.
- [Adelson and Bergen, 1985] Edward H. Adelson and James R. Bergen. Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America A*, 2(2) :284–299, 1985.
- [Adelson, 2003] Beth Adelson. Bringing considerations of situated action to bear on the paradigm of cognitive modeling : the 2002 benjamin franklin medal in computer and cognitive science presented to lucy suchman. *Journal of the Franklin Institute*, 340 :283–292, 2003.
- [Alecú and Frezza-Buet, 2008] Lucian Alecú and Hervé Frezza-Buet. Are neural fields suitable for vector quantization ? In *Proceedings of The Seventh International Conference on Machine Learning and Applications (ICMLA '08)*, pages 239–244, San Diego (CA, USA), December 2008. IEEE.
- [Alecú and Frezza-Buet, 2009a] Lucian Alecú and Hervé Frezza-Buet. Reconciling neural fields to self-organization. In Michel Verleysen, editor, *European Symposium on Artificial Neural Networks - Advances in Computational Intelligence and Learning (ESANN)*, Bruges, Belgium, pages 571–576, April 2009.
- [Alecú and Frezza-Buet, 2009b] Lucian Alecú and Hervé Frezza-Buet. Application-driven parameter tuning methodology for dynamic neural field equations. In *Neural Information Processing, ICONIP'09 Proceedings, Part I*, volume 5863/2009 of *Lecture Notes in Computer Science*, pages 135–142, Bangkok (Thailand), 2009. Springer Berlin / Heidelberg.
- [Alecú *et al.*, 2011] Lucian Alecú, Hervé Frezza-Buet, and Frédéric Alexandre. Can self-organization emerge through dynamic neural fields computation ? . *Connection Science*, 23(1) :1–31, 2011.
- [Alecú, 2005] Lucian Alecú. Architecture générique de vidéo-surveillance multi-capteurs. Technical report, Université Polytechnique de Bucarest, 2005. Mémoire de fin d'études.
- [Alexandre *et al.*, 1991] F. Alexandre, F. Guyot, J.-P. Haton, and Y. Burnod. The Cortical Column : A New Processing Unit for Multilayered Networks. *Neural Networks*, 4 :15–25, 1991.
- [Amari, 1977] Shun-ichi Amari. Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological Cybernetics*, 27 :77–87, 1977.
- [Ananthanarayanan and Modha, 2007] Rajagopal Ananthanarayanan and Dharmendra Modha. Anatomy of a cortical simulator. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC07)*, 2007.

- [Andry *et al.*, 2001] Pierre Andry, Philippe Gaussier, Sorin Moga, Jean Paul Banquet, and Jacqueline Nadel. Learning and communication via imitation : An autonomous robot perspective. *IEEE Transactions on Systems, Man and Cybernetics, part A : Systems and Humans*, 41(5) :431–442, 2001.
- [Aubert and Kornprobst, 2006] Gilles Aubert and Pierre Kornprobst. *Mathematical problems in image processing : Partial Differential Equations and the Calculus of Variations*, volume 147 of *Applied Mathematical Sciences*. Springer, 2nd edition, 2006.
- [Ballard, 1986] D. H. Ballard. Cortical connections and parallel processing : Structure and function. *Behavioral Brain Science*, 9 :67–129, 1986.
- [Baraduc and Guigon, 2002] Pierre Baraduc and Emmanuel Guigon. Population computation of vectorial transformations. *Neural Computation*, 14(4) :845–871, 2002.
- [Baraduc *et al.*, 2001] Pierre Baraduc, Emmanuel Guigon, and Yves Burnod. Recoding arm position to learn visuomotor transformations. *Cerebral Cortex*, 11(10) :906–917, 2001.
- [Baraduc, 1999] Pierre Baraduc. *Modèle neuronal des transformations de coordonnées. Contrôle visiomoteur par recodage de la proprioception*. PhD thesis, Université de Paris 06, 1999.
- [Barreto and Souza, 2006] Guilherme A. Barreto and Luís Gustavo M. Souza. Adaptive filtering with the self-organizing map : A performance comparison. *Neural Networks*, 19(6-7) :785–798, 2006.
- [Barreto *et al.*, 2003] Guilherme A. Barreto, Aluizio F. R. Araújo, and Helge J. Ritter. Self-organizing feature maps for modeling and control of robotic manipulators. *Journal of Intelligent and Robotic Systems*, 36(4) :407–450, 2003.
- [Beer, 2000] Randall D. Beer. Dynamical approaches to cognitive science. *Trends in Cognitive Sciences*, 4(3) :91–99, 2000.
- [Beer, 2003] Randall D. Beer. The dynamics of active categorical perception in an evolved model agent. *Adaptive Behavior*, 11(4) :209–243, 2003.
- [Beer, 2006] Randall D. Beer. Beyond control : The dynamics of brain-body-environment interaction in motor systems. In D. Sternad, editor, *Progress in Motor Control V : A Multidisciplinary perspective*. Springer, 2006. to appear.
- [Bergener *et al.*, 1999] T. Bergener, C. Bruckhoff, P. Dahm, H. Janßen, F. Joublin, R. Menzner, and A. Steinhage. Complex behavior by means of dynamical systems for an anthropomorphic robot. *Neural Networks*, 12 :1087–1099, 1999.
- [Berry and Temam, 2007] Hugues Berry and Olivier Temam. Modeling self-developing biological neural networks. *Neurocomputing*, 70 :2723–2734, 2007.
- [Berthoz, 1997] Alain Berthoz. *Le sens du mouvement*. Sciences. Odile Jacob, 1997.
- [Binzegger *et al.*, 2005] Tom Binzegger, Rodney J. Douglas, and Kevan A. C. Martin. Cortical architecture. In M. De Gregorio, V. Di Maio, M. Frucci, and C. Musio, editors, *Brain, Vision, and Artificial Intelligence*, LNCS 3704, pages 15–28. Springer-Verlag, 2005.
- [Bishop, 2004] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford university press, 2004.
- [Bishop, 2006] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Spring, 2006.
- [Blanchet and Alexandre, 1995] P. Blanchet and F. Alexandre. A Hierarchical Connectionist Architecture for Learning Internal Representations of Behaviours. In *Proceedings International Conference on Artificial Neural Networks*, volume 2, pages 183–188, Paris, October 1995.

-
- [Blanchet, 1992] Pascal Blanchet. *Une architecture connexionniste pour l'apprentissage par l'expérience et la représentation des connaissances*. PhD thesis, Université Paris XI Orsay, 1992.
- [Blanchet, 1994] P. Blanchet. An Architecture for Representing and Learning Behaviors by Trial and Error. In *Proceedings Congress SAB'94 : from Animals to Animats III, Simulation of Adaptive Behavior*, Brighton (England), August 1994.
- [Bressloff and Folias, 2004] Paul C. Bressloff and Stefanos E. Folias. Front bifurcations in an excitatory neural network. *SIAM Journal of Applied Mathematics*, 65(1) :131–151, 2004.
- [Bressloff et al., 2002] Paul C. Bressloff, Jack D. Cowan, Martin Golubitsky, Peter J. Thomas, and Matthew C. Wiener. What geometric visual hallucinations tell us about the visual cortex. *Neural Computation*, 14 :473–491, 2002.
- [Bressloff, 2005] Paul C. Bressloff. Spontaneous symmetry breaking in self-organizing neural fields. *Biological Cybernetics*, 93 :256–274, 2005.
- [Brette et al., 2007] Romain Brette, Michelle Rudolph, Ted Carnevale, Michael Hines, David Beeman, James M. Bower, Markus Diesmann, Abigail Morrison, Philip H. Goodman, Frederick C. Harris, Jr., Milind Zirpe, Thomas Natschläger, Dejan Pecevski, Bard Ermentrout, Mikael Djurfeldt, Anders Lansner, Olivier Rochel, Thierry Vieville, Eilif Muller, Andrew P. Davison, Sami El Boustani, and Alain Destexhe. Simulation of networks of spiking neurons : a review of tools and strategies. *Journal of Computational Neuroscience*, 23(3) :349–398, 2007.
- [Brooks and Stein, 1994] R. A. Brooks and L. A. Stein. Building brains for bodies. *Autonomous Robot*, 1(1) :7–25, 1994.
- [Brooks, 1990] Rodney A. Brooks. Elephants don't play chess. *Robotics and Autonomous Systems*, 6(1-2) :3–15, 1990.
- [Brooks, 1994] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1), 1994.
- [Bullier, 2001] Jean Bullier. Integrated model of visual processing. *Brain Research Reviews*, 36 :96–107, 2001.
- [Burnod et al., 1999] Yves Burnod, Pierre Baraduc, Alexandra Battaglia-Mayer, Emmanuel Guigon, Étienne Koechlin, Stefano Ferraina, Francesco Lacquaniti, and Roberto Caminiti. Parieto-frontal coding of reaching : An integrated framework. *Exp Brain Res*, 129(3) :325–346, 1999.
- [Burnod, 1989] Yves Burnod. *An Adaptive Neural Network : The Cerebral Cortex*. Masson, 1989.
- [Carretero, 2007] Jérôme Carretero. Plate-forme de comparaison de champs de neurones dynamiques. Master's thesis, Université Henri Poincaré - Nancy I, 2007.
- [Chua and Yang, 1988] Leon O. Chua and L. Yang. Cellular neural networks : Theory. *IEEE Transactions on Circuits and Systems*, 35 :1257–1272, 1988.
- [Church, 1936] Alonzo Church. An unsolvable problem of elementary number theory. *American J of Math.*, 58 :345–363, 1936.
- [Clancey, 2006] William J. Clancey. Scientific antecedents of situated cognition. In Phillips Robins and Murat Aydede, editors, *Cambridge Handbook of Situated Cognition*, chapter 2. Cambridge University Press, 2006.
- [Clark and Grush, 1999] Andy Clark and Rick Grush. Towards a cognitive robotics. *Adaptive Behavior*, 7(1) :5–16, 1999.

- [Clark, 1998a] Andy Clark. Embodiment and the philosophy of mind. In *Current Issues In Philosophy Of Mind : Royal Institute Of Philosophy Supplement*, volume 43, pages 35–52. Cambridge University Press :, 1998.
- [Clark, 1998b] Andy Clark. Where brain, body and world collide. *Daedalus : Journal of the American Academy of Arts and Sciences*, 127(2) :257–280, 1998. Special Issue on The Brain.
- [Clark, 2002] Andy Clark. Is seeing all it seems? action, reason and the grand illusion. *Journal of Consciousness Studies*, 9(5-6), 2002.
- [Clark, 2003] Andy Clark. Artificial intelligence and the many faces of reason. In S. P. Stich and T. A. Warfield, editors, *The Blackwell guide to philosophy and mind*. Blackwell, 2003.
- [Coore, 2005] Daniel Coore. Introduction to amorphous computing. In *Unconventional Programming Paradigms*, volume 3566 of *Lecture Notes in Computer Science*, pages 99–109. Springer, 2005.
- [Crook, 2007] Nigel Crook. Nonlinear transient computation. *Neurocomputing*, 70 :1167–1176, 2007.
- [CSIM, 2008] CSIM, 2008. <http://www.csim.com>, Consulté le 15 janvier 2008.
- [Cummings, 1995] Jeffrey L. Cummings. Anatomic and behavioral aspects of frontal-subcortical circuits. In Jordan Grafman, Keith J. Holyoak, and François Boller, editors, *Structure and Functions of the Human Prefrontal Cortex*, volume 769, pages 1–13. Annals of the New York Academy of Sciences, 1995.
- [Damasio, 1995] Antonio R. Damasio. On some functions of the human prefrontal cortex. In Jordan Grafman, Keith J. Holyoak, and François Boller, editors, *Structure and functions of the human prefrontal cortex*, volume 769, pages 241–251. Annals of the New York Academy of Sciences, 1995.
- [Dauce et al., 1998] Emmanuel Dauce, Mathias Quoy, Bruno Cessac, Bernard Doyon, and Manuel Samuelides. Self-organization and dynamics reduction in recurrent networks : stimulus presentation and learning. *Neural Networks*, 11 :521–533, 1998.
- [Daugmann, 1985] John G. Daugmann. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of Optical Society of America A*, 2(7) :1160–1169, 1985.
- [David and Friston, 2003] Olivier David and Karl J. Friston. A neural mass model for meg/eeg : coupling and neuronal dynamics. *NeuroImage*, 20 :1743–1755, 2003.
- [Deguet et al., 2006] J. Deguet, Y. Demazeau, and L. Magnin. Elements about the emergence issue : A survey of emergence definitions. *Complexus*, 3 :24–31, 2006.
- [Dehaene and Changeux, 1989] Stanislas Dehaene and Jean-Pierre Changeux. A simple model of prefrontal cortex function in delayed-response task. *Journal of Cognitive Neuroscience*, 1(3) :244–261, 1989.
- [DeHon, 2002] André DeHon. Very large scale spatial computing. In *Proceedings of the Third International Conference on Unconventional Models of Computation (UMC'02)*, pages 27–37, 2002.
- [Deneve et al., 1999] S. Deneve, P. E. Latham, and A. Poujet. Reading population codes : a neural implementation of ideal observers. *Nature Neuroscience*, 2(8) :740–745, 1999.
- [Dennett, 1993] Daniel C. Dennett. Review of f. varela, e. thompson and e. rosch, the embodied mind. *American Journal of Psychology*, 106 :121–126, 1993.

-
- [Diks and Panchenko, 2006] Cees Diks and Valentyn Panchenko. A new statistic and practical guidelines for nonparametric granger causality testing. *Journal of Economic Dynamics & Control*, 30 :1647–1669, 2006.
- [D’Innocenzo *et al.*, 2006] A. D’Innocenzo, F. Paladini, and L. Renna. Leaky tap behavior described by a simple discrete mapping. *Nonlinear Dynamics*, 43 :379–390, 2006.
- [Djurfeldt *et al.*, 2008] M. Djurfeldt, M. Lundqvist, C. Johansson, M. Rehn, Ö. Ekeberg, and A. Lansner. Brain-scale simulation of the neocortex on the ibm blue gene/l supercomputer. *IMB Journal of Research and Development*, 52(1-2), 2008.
- [Dominey, 1995] Peter Ford Dominey. Complex sensory-motor sequence learning based on recurrent state representation and reinforcement learning. *Biological Cybernetics*, 73 :265–274, 1995.
- [Dor, 1985] Joël Dor. *Introduction à la lecture de Lacan*, volume 1. L’inconscient structuré comme un langage. Denoël, 1985.
- [Douglas and Martin, 2007] Rodney J. Douglas and Kevan A.C. Martin. Recurrent neuronal circuits in the neocortex. *Current Biology*, 17(13) :R496–R500, 2007.
- [Douvier, 2003] Yann-Éric Douvier. évaluation asynchrone des éléments d’un champ de neurones corticaux. Master’s thesis, Université Henri Poincaré - Nancy I, 2003.
- [Doya, 1999] Kenji Doya. What are the computations of the cerebellum, the basal ganglia and the cerebral cortex? *Neural Networks*, 12(7-8) :961–974, 1999.
- [Drumea and Frezza-Buet, 2007] Georges Adrian Drumea and Hervé Frezza-Buet. Tracking fast changing non-stationary distributions with a topologically adaptive neural network : application to video tracking. In *ESANN, European Symposium on Artificial Neural Networks*, 2007. 6 pages.
- [Drumea, 2006] George Adrian Drumea. Experimenting with growing neural gas on a video surveillance platform. Technical report, Université Polytechnique de Bucarest, 2006. Mémoire de fin d’études.
- [Duquaire, 2003] Paul-Victor Duquaire. Introduction à la pensée de francisco j. varela à partir de autonomie et connaissance et l’inscription corporelle de l’esprit. *Les Cahiers de l’ATP*, 2003.
- [Durand and Alexandre, 1996] Stéphane Durand and Frédéric Alexandre. Tom, a new temporal neural net architecture for speech signal processing. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP’96)*, volume 6, pages 3549–3552, 1996.
- [Elbert and Rockstroh, 2004] Thomas Elbert and Brigitte Rockstroh. Reorganization of human cerebral cortex : The range of changes following use and injury. *The Neuroscientist*, 10(2) :129–141, 2004.
- [Elman, 1990] J. L. Elman. Finding structure in time. *Cognitive Science*, 14 :179–211, 1990.
- [Enculescu and Bestehorn, 2007] Mihaela Enculescu and Michael Bestehorn. Mechanisms for activity spread in a neural field model. *Neurocomputing*, 70 :2996–3000, 2007.
- [Felleman and Essen, 1991] Daniel J. Felleman and David C. Van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex*, 1(1) :1–47, 1991.
- [Fellenz and Taylor, 2002] Winfried A. Fellenz and John G. Taylor. Establishing retinotopy by lateral-inhibition type homogeneous neural fields. *Neurocomputing*, 48 :313–322, 2002.

- [Fix *et al.*, 2007] Jeremy Fix, Nicolas Rougier, and Frederic Alexandre. From physiological principles to computational models of the cortex. *Journal of Physiology-Paris*, 101(1-3) :32–39, 2007.
- [Folias and Bressloff, 2004] Stefanos E. Folias and Paul C. Bressloff. Breathing pulses in an excitatory neural network. *SIAM Journal of Applied Dynamical Systems*, 3(3) :378–407, 2004.
- [Folias and Bressloff, 2005] Stefanos E. Folias and Paul C. Bressloff. Breathers in two-dimensional neural media. *Physical Review Letters*, 95, 2005.
- [Folias, 2008] Stefanos E. Folias, 2008. <http://math.bu.edu/people/sf/breathers.html>, consulté le 2 janvier 2008.
- [Fressengeas and Frezza-Buet, 2006] Nicolas Fressengeas and Hervé Frezza-Buet. Generic method for solving partial differential equations through the design of problem-specific cellular neural networks, 2006. arXiv :math-ph/0610037v5.
- [Frezza-Buet and Alexandre, 1999] Hervé Frezza-Buet and Frédéric Alexandre. Specialization within cortical models : An application to causality learning. In *Proceedings of the 7th European Symposium on Artificial Neural Networks (ESANN)*, 1999. 6 pages.
- [Frezza-Buet and Alexandre, 2002] Hervé Frezza-Buet and Frédéric Alexandre. From a biological to a computational model for the autonomous behavior of an animat. *Information Sciences*, 144(1-4) :1–43, 2002.
- [Frezza-Buet and Ménard, 2005] Hervé Frezza-Buet and Olivier Ménard. Making competition in neural fields suitable for computational architectures. In Wlodzislaw Duch, Janusz Kacprzyk, and Erkki Oja, editors, *International Conference on Artificial Neural Networks (ICANN)*, Lecture Notes in Computer Science (LNCS) 3696, part I, pages 217–222. Springer-Verlag, 2005.
- [Frezza-Buet *et al.*, 2001] Hervé Frezza-Buet, Nicolas Rougier, and Frédéric Alexandre. Integration of biologically inspired temporal mechanisms into a cortical framework for sequence processing. In Ron Sun and Lee C. Giles, editors, *Sequence Learning : Paradigms, Algorithms, and Applications*, Lecture Notes in Computer Science (LNCS) 1828, pages 321–348. Springer-Verlag, 2001.
- [Frezza-Buet, 1999] Hervé Frezza-Buet. *Un modèle de cortex pour le comportement motivé d'un agent neuromimétique autonome*. PhD thesis, Université Henri Poincaré - Nancy I, oct 1999. In french.
- [Frezza-Buet, 2002] Hervé Frezza-Buet. Action scheme scheduling with a neural architecture : A prefrontal cortex approach. In J.R. Dorronsoro, editor, *International Conference on Artificial Neural Networks (ICANN)*, Lecture Notes in Computer Science (LNCS) 2415, pages 271–276. Springer-Verlag, 2002.
- [Frezza-Buet, 2008] Hervé Frezza-Buet. Following non-stationary distributions by controlling the vector quantization accuracy of a growing neural gas network. *Neurocomputing*, 71(7-9) :1191–1202, 2008. extended version of *Tracking fast changing non-stationary distributions with a topologically adaptive neural network : application to video tracking*, ESANN'07.
- [Friston, 2002] Karl Friston. Functional integration and inference in the brain. *Progress in Neurobiology*, 68 :113–143, 2002.
- [Fritzke, 1995] B. Fritzke. A growing neural gas network learns topologies. In G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 625–632. MIT Press, Cambridge MA, 1995.

-
- [Fritzke, 1997a] B. Fritzke. The LBG-U method for vector quantization – an improvement over LBG inspired from neural networks. *Neural Processing Letters*, 5(1) :35–45, 1997.
- [Fritzke, 1997b] B. Fritzke. A self-organizing network that can follow non-stationary distributions. In *ICANN'97 : International Conference on Artificial Neural Networks*, pages 613–618. Springer, 1997.
- [Fuster, 1997] Joaquin M.D Fuster. *The Prefrontal Cortex : Anatomy, Physiology, and Neuropsychology of the Frontal Lobe*. Lippincott Williams and Wilkins Publishers, third edition, 1997.
- [Fuster, 2000] Joaquin M. Fuster. Cortical dynamics of memory. *International Journal of Psychophysiology*, 35(2-3) :155–164, 2000.
- [Gallese and Lakoff, 2005] Vittorio Gallese and George Lakoff. The brain’s concepts : the role of the sensory-motor system in conceptual knowledge. *Cognitive Neuropsychology*, 22(3-4) :455–479, 2005.
- [Gaussier *et al.*, 1998] P. Gaussier, S. Moga, J.-P. Banquet, and M. Quoy. From perception-action loops to imitation processes. *Applied Artificial Intelligence*, 1(7) :701–727, 1998.
- [Gaussier *et al.*, 2000] P. Gaussier, C. Joulain, J.P. Banquet, S. Leprêtre, and A. Revel. The visual homing problem : an example of robotics/biology cross fertilization. *Robotics and Autonomous Systems*, 30 :155–180, 2000.
- [Gaussier *et al.*, 2001] P. Gaussier, A. Revel, J.-P. Banquet, and V. Babeau. From view cells and place cells to cognitive map learning : processing stages of the hippocampal system. *Biological Cybernetics*, 86(1) :15–28, 2001.
- [Gaussier *et al.*, 2002] Philippe Gaussier, Arnaud Revel, Jean-Paul Banquet, and V. Babeau. From view cells and place cells to cognitive map learning : processing stages of the hippocampal system. *Biological Cybernetics*, 86 :15–28, 2002.
- [GENESIS, 2008] GENESIS, 2008. <http://www.genesis-sim.org>, Consulté le 4 janvier 2008.
- [Gerstner and Kistler, 2002] Wulfram Gerstner and Werner M. Kistler. *Spiking Neuron Models Single Neurons, Populations, Plasticity*. Cambridge University Press, 2002.
- [Gerstner, 1995] Wulfram Gerstner. Time structure of the activity in neural network models. *Physical Review E*, 51 :738–758, 1995.
- [Ghițescu, 2004] Anca Ghițescu. Adding events management in a general-purpose neural network simulator. Technical report, Université Polytechnique de Bucarest, 2004. Mémoire de fin d’études.
- [Gibson, 1979] J. J. Gibson. *The Ecological Approach to Visual Perception*. Houghton Mifflin, 1979. Currently published by Lawrence Erlbaum.
- [Giese, 2003] Martin A. Giese. Learning recurrent neural models with minimal complexity from neural tuning data. *Neurocomputing*, 52 :277–283, 2003.
- [Girau, 2007] Bernard Girau. *Du parallélisme connexionniste à une pratique de calcul distribué numérique bio-inspiré*. PhD thesis, Université Nancy II, 2007.
- [Gleick, 1991] James Gleick. *La théorie du chaos, vers une nouvelle science*. Champs. Flammarion, 1991. Titre original : *Chaos : Making a New Science* écrit en 1987.
- [Goh and Crook, 2007] Wee Jin Goh and Nigel Crook. Pattern recognition using chaotic transients. In *European Symposium on Artificial Neural Networks (ESANN'2007)*, 2007.
- [Goldberg, 1989] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.

- [Gourévitch *et al.*, 2006] Boris Gourévitch, Régine Le Bouquin-Jeannès, and Gérard Faucon. Linear and nonlinear causality between signals : methods, examples and neurophysiological applications. *Biological Cybernetics*, 95 :349–369, 2006.
- [Grimbert and Faugeras, 2006] François Grimberty and Olivier Faugeras. Bifurcation analysis of jansen’s neural mass model. *Neural Computation*, 18 :3052–3068, 2006.
- [Grossberg and Schmajuk, 1987] Stephen Grossberg and Nestor Schmajuk. Neural dynamics of attentionally modulated pavlovian conditioning : Conditioned reinforcement, inhibition, and opponent processing. *Psychology*, 15(3) :195–240, 1987.
- [Grossberg and Schmajuk, 1989] Stephen Grossberg and Nestor Schmajuk. Neural dynamics of adaptive timing and temporal discrimination during associative learning. *Neural Networks*, 2 :79–102, 1989.
- [Grossberg, 1987] Stephen Grossberg. Competitive learning : From interactive activation to adaptive resonance. *Cognitive Science*, 11 :23–63, 1987.
- [Grossberg, 1999] Stephen Grossberg. How does the cerebral cortex work? learning, attention and grouping by the laminar circuits of visual cortex. *Spatial Vision*, 12 :163–186, 1999.
- [Grossberg, 2003] Stephen Grossberg. How does the cerebral cortex work? development, learning, attention, and 3d vision by laminar circuits of visual cortex. *Behavioral and Cognitive Neuroscience Reviews*, 2 :47–76, 2003.
- [Gruau, 1993] Frédéric Gruau. Adding learning to cellular development of neural networks : Evolution and the baldwin effect. *Evolutionary Computation*, 1(3) :213–233, 1993.
- [Gruau, 1995] Frédéric Gruau. Automatic definition of modular neural networks. *Adaptive Behavior*, 3(2) :151–183, 1995.
- [Guigon, 1993] Emmanuel Guigon. *Modélisation des propriétés du cortex cérébral - Comparaison entre aires visuelles, motrices et préfrontales*. PhD thesis, École Centrale de Paris, 1993.
- [Gutowitz and Langton, 1988] H. A. Gutowitz and C. G. Langton. Methods for designing cellular automata with ”interesting” behavior. *CNLS News Letter*, 1988.
- [Gutzwiller *et al.*, 2010] Jean-Louis Gutzwiller, Hervé Frezza-Buet, and Olivier Pietquin. Online Speaker Diarization with a Size-Monitored Growing Neural Gas Algorithm. In Michel Verleysen, editor, *Proceedings of the 18 th European Symposium on Artificial Neural Networks (ESANN), Computational Intelligence and Machine Learning*, pages 505–510, Bruges (Belgium), April 2010.
- [Harnad, 1990] Stevan Harnad. The symbol grounding problem. *Physica D*, 42 :335–346, 1990.
- [Haselager *et al.*, 2003] W. F. G. Haselager, R. M. Bongers, and I. van Rooij. Cognitive science, representations and dynamical systems theory. In W. Tschacher and J.-P. Dauwalder, editors, *The dynamical systems approach to cognition : Concepts and empirical paradigms based on self-organization, embodiment, and coordination Dynamics.*, volume 10 of *Studies of Nonlinear Phenomena in Life Science*, pages 229–242. World Scientific, 2003.
- [Haselager, 1999] W. F. G. Haselager. Neurodynamics and the revival of associationism in cognitive science. In A. Riegler, M. Peschl, and A. Von Stein, editors, *Understanding representation in the cognitive sciences : Does representation need reality ?*, pages 115–120. New York : Kluwer Academic / Plenum, 1999.
- [Hauk *et al.*, 2004] Olaf Hauk, Ingrid Johnsrude, and Friedemann Pulvermüller. Somatotopic representation of action words in human motor and premotor cortex. *Neuron*, 41(2) :301–307, 2004.

-
- [Henri, 2005] Jacques Henri. Auto-organisation visuelle corticale située : prise en compte de la rétinitopie. Master's thesis, Université Henri Poincaré - Nancy I, 2005.
- [Henrya *et al.*, 2007] Frédéric Henrya, Emmanuel Dauce, and Hédi Soula. Temporal pattern identification using spike-timing dependent plasticity. *Neurocomputing*, 70 :2009–2016, 2007.
- [Hertz *et al.*, 1991] John Hertz, Anders Krogh, and Richard G. Palmer. *Introduction to the Theory of Neural Computation*. Addison-Wesley, 1991.
- [Hodgkin and Huxley, 1952] A. Hodgkin and A. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology*, 117 :500–544, 1952.
- [Hopfield and Brody, 2000] J. J. Hopfield and Carlos D. Brody. What is a moment ? "cortical" sensory integration over a brief interval. *PNAS*, 97 :13919—13924, 2000.
- [Hopfield and Brody, 2001] J. J. Hopfield and Carlos D. Brody. What is a moment ? transient synchrony as a collective mechanism for spatiotemporal integration. *PNAS*, 98 :1282—1287, 2001.
- [Hopfield, 1982] John J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the USA*, 79(8) :2554–2558, 1982.
- [Horta and Erlhagen, 2006] Claudia Horta and Wolfram Erlhagen. Robust persistent activity in neural fields with asymmetric connectivity. *Neurocomputing*, 69 :1141–1145, 2006.
- [Hubel, 1994] David Hubel. *L'Œuil, le Cerveau et la Vision : les étapes cérébrales du traitement visuel*. L'Univers des Sciences. Pour la science, diffusion Belin, 1994.
- [Hupe *et al.*, 1998] J. M. Hupe, A. C. James, B. R. Payne, S. G. Lomber, P. Girard, and J. Bullier. Cortical feedback improves discrimination between figure and background by v1, v2 and v3 neurons. *Nature*, 394(6695) :784–787, 1998.
- [Hutt *et al.*, 2003] A. Hutt, M. Bestehorn, and T. Wennekers. Pattern formation in intracortical neuronal fields. *Network : Computation in Neural Systems*, 14 :351–368, 2003.
- [Igel *et al.*, 2001] Christian Igel, Wolfram Erlhagen, and Dirk Jancke. Optimization of dynamic neural fields. *Neurocomputing*, 36 :225–233, 2001.
- [Iarraza-Lomelí *et al.*, 1999] Aquiles Iarraza-Lomelí C. M. Arizmendi, and A. L. Salas-Brito. Controlling a leaky tap. *Physics letters. A*, 259(2) :115–128, 1999.
- [InterCell, 2008] InterCell, 2008. <http://intercell.metz.supelec.fr>.
- [Iossifidis and Steinhage, 2001] I. Iossifidis and A. Steinhage. Controlling an 8 dof manipulator by means of neural fields. In A. Halme, R. Chatila, and E. Prassler, editors, *Proceedings of the International Conference on Field and Service Robotics (FSR 2001)*, Helsinki, 2001.
- [Izhikevich and Desai, 2003] Eugene M. Izhikevich and Niraj S. Desai. Relating stdp to bcm. *Neural Computation*, 15 :1511–1523, 2003.
- [Izhikevich and FitzHugh, 2006] M. Izhikevich and Richard FitzHugh. Fitzhugh-nagumo model, 2006. http://www.scholarpedia.org/article/FitzHugh-Nagumo_Model, consulté le 16 décembre 2007.
- [Izhikevich *et al.*, 2003] Eugene M. Izhikevich, Niraj S. Desai, Elisabeth C. Walcott, and Frank C. Hoppensteadt. Bursts as a unit of neural information : selective communication via resonance. *Trends in Neurosciences*, 26(3) :161–167, 2003.
- [Izhikevich, 2002] Eugene M. Izhikevich. Resonance and selective communication via bursts in neurons having subthreshold oscillations. *Biosystems*, 67(1-3) :95–102, 2002.

- [Izhikevich, 2004] Eugene M. Izhikevich. Which model to use for cortical spiking neurons? *IEEE Transactions on Neural Networks*, 15 :1063–1070, 2004.
- [Izhikevich, 2006] Eugene M. Izhikevich. Polychronization : Computation with spikes. *Neural Computation*, 18(2) :245–282, 2006.
- [Jaeger and Haas, 2004] Herbert Jaeger and Harald Haas. Harnessing nonlinearity : Predicting chaotic systems and saving energy in wireless communication. *Science*, 304(5667) :78–80, 2004.
- [Jansen and Rit, 1995] Ben H. Jansen and Vincent G. Rit. Electroencephalogram and visual evoked potential generation in a mathematical model of coupled cortical columns. *Biological Cybernetics*, 73(4) :357–366, 1995.
- [Jansen *et al.*, 1995] Ben H. Jansen, George Zouridakis, and Michael E. Brandt. A neurophysiologically-based mathematical model of flash visual evoked potentials. *Biological Cybernetics*, 68(3) :275—283, 1995.
- [JFPDA, 2008] JFPDA, 2008. <http://www.metz.supelec.fr/JFPDA2008>.
- [Jones, 2000] Edward G. Jones. Microcolumns in the cerebral cortex. *PNAS*, 97(10) :5019–5021, 2000.
- [Jordan, 1986] Michael I. Jordan. Attractor dynamics and parallelism in a connectionist sequential machine. In *Proceedings of the eighth Annual Conference of the Cognitive Science Society*. Erlbaum, 1986.
- [Keerthi *et al.*, 1999] S. S. Keerthi, S. K. Shevade, . Bhattacharyya, and K. R. K. Murthy. Improvements to platt’s smo algorithm for svm classifier design. Technical Report CD-99-14, National University of Singapore, 1999.
- [Kish, 2002] Laszlo B. Kish. End of moore’s law : thermal (noise) death of integration in micro and nano electronics. *Physics Letters A*, 305 :144–149, 2002.
- [Knight, 1972] Bruce W. Knight. Dynamics of encoding in a population of neurons. *The Journal of General Physiology*, 59 :734–766, 1972.
- [Kohonen, 1997] Teuvo Kohonen. *Self Organizing Maps*. Springer, 1997. Second Edition.
- [Kozek *et al.*, 1995] Tibor Kozek, Leon O. Chua, Tamás Roska, Dietrich Wolf, Ronald Tetzlaff, Frank Puffer, and Károly Lotz. Simulating nonlinear waves and partial differential equations via cnn-part ii : Typical examples. *IEEE Transactions on Circuits and Systems-I : Fundamental theory and applications*, 42(10) :807–815, october 1995.
- [Kubík, 2003] Aleš Kubík. Toward a formalization of emergence. *Artif. Life*, 9(1) :41–65, 2003.
- [Lacan, 1973] Jacques Lacan. *Les quatre concepts fondamentaux de la psychanalyse*. Points, Essais. Editions du Seuil, 1973. Le séminaire (1964), livre XI, texte établi par Jacques-Alain Miller.
- [Lacan, 1978] Jacques Lacan. *Le moi dans la théorie de Freud et dans la technique de la psychanalyse*. Points, Essais. Editions du Seuil, 1978. Le séminaire (1954-1955), livre II, texte établi par Jacques-Alain Miller.
- [Langton, 1990] C. G. Langton. Computation at the edge of chaos : Phase transitions and emergent computation. *Physica D*, 42 :12–37, 1990.
- [Legenstein and Maass, 2007a] Robert Legenstein and Wolfgang Maass. Edge of chaos and prediction of computational performance for neural circuit models. *Neural Networks*, 20(3) :323–334, 2007.

-
- [Legenstein and Maass, 2007b] Robert Legenstein and Wolfgang Maass. What makes a dynamical system computationally powerful? In S. Haykin, J.C. Principe, T. Sejnowski, and J. McWhirter, editors, *New directions in statistical signal processing : From systems to brain*,, pages 127–154. MIT Press, 2007.
- [Legrand, 2006] Thomas Legrand. Apprentissage par renforcement dans un système sensorimoteur d’inspiration corticale. Master’s thesis, Université Henri Poincaré - Nancy I, 2006.
- [Lloyd, 2000] Seth Lloyd. Ultimate physical limits to computation. *Nature*, 406 :1047–1054, 2000.
- [Lorenz, 1963] Edward N. Lorenz. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20 :130–141, 1963.
- [Lücke and Bouecke, 2005a] J. Lücke and J. D. Bouecke. Dynamics of cortical columns - self-organization of receptive fields. In *Proc. ICANN*, LNCS 3696, pages 31–37. Springer, 2005.
- [Lücke and Bouecke, 2005b] Jörg Lücke and Jan D. Bouecke. Dynamics of cortical columns - self organization of receptive fields. In Włodzisław Duch, Janusz Kacprzyk, and Erkki Oja, editors, *ICANN 2005*, number 3696 in Lecture Notes in Computer Science. Springer-Verlag, 2005.
- [Lücke, 2005] J. Lücke. Dynamics of cortical columns - sensitive decision making. In *Proc. ICANN*, LNCS 3696, pages 25 – 30. Springer, 2005.
- [Lukoševičius and Jaeger, 2009] Mantas Lukoševičius and Herbert Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3) :127–149, 2009.
- [Maass and Markram, 2004] Wolfgang Maass and Henry Markram. On the computational power of circuits of spiking neurons. *Journal of Computer and System Sciences*, 69(4) :593–616, 2004.
- [Maass et al., 2002] Wolfgang Maass, Thomas Natschläger, and Henry Markram. Real-time computing without stable states : A new framework for neural computation based on perturbations. *Neural Computation*, 14 :2531–2560, 2002.
- [Martinez and Schulten, 1994] T. M. Martinez and K. J. Schulten. Topology representing networks. *Neural Networks*, 7(3) :507–522, 1994.
- [McCarthy and Hayes, 1969] J. McCarthy and P. J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In D. Michie and B. Meltzer, editors, *Machine Intelligence*, volume 4, pages 463–502. Edinburgh University Press, 1969.
- [McCulloch and Pitts, 1943] W. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 7 :115–133, 1943.
- [McGurk and MacDonald, 1976] Harry McGurk and John MacDonald. Hearing lips and seeing voices. *Nature*, 264 :746–748, 1976.
- [Meiss, 2007] James Meiss. Dynamical systems, 2007. http://www.scholarpedia.org/article/Dynamical_Systems, consulté le 11 novembre 2007.
- [Ménard and Frezza-Buet, 2005] Olivier Ménard and Hervé Frezza-Buet. Model of multi-modal cortical processing : Coherent learning in self-organizing modules. *Neural Networks*, 18(5-6) :646–655, 2005. extended version of *Coherent learning in cortical maps : A generic approach*, IJCNN’05.
- [Ménard, 2006] Olivier Ménard. *Mécanismes d’inspiration corticale pour l’apprentissage et la représentation d’asservissements sensori-moteurs en robotique*. PhD thesis, Université Henri Poincaré, 2006.

- [Menzner *et al.*, 2000] R. Menzner, A. Steinhage, and W. Erlhagen. Generating interactive robot behavior : A mathematical approach. In *From Animals to Animats 5 : Proceedings of the Sixth International Conference on Simulation of Adaptive Behavior*, 2000.
- [Merleau-Ponty, 1964] Maurice Merleau-Ponty. *Le visible et l'invisible*. Galimard, 1964.
- [Miikkulainen *et al.*, 2005] Risto Miikkulainen, James A. Bednar, Yoonsock Choe, and Joseph Sirosh. *Computational Maps in the Visual Cortex*. Springer, 2005.
- [Mikhailova and Goerick, 2005] Inna Mikhailova and Christian Goerick. Conditions of activity bubble uniqueness in dynamic neural fields. *Biological cybernetics*, 92(2) :82–91, 2005.
- [Miller *et al.*, 2001] K. D. Miller, D. J. Simons, and D. J. Pinto. Processing in layer 4 of the neocortical circuit : New insights from visual and somatosensory cortex. *Current Opinion in Neurobiology*, 11 :488–497, 2001.
- [Mingolla *et al.*, 1999] E. Mingolla, W. Ross, and S. Grossberg. A neural network for enhancing boundaries and surfaces in synthetic aperture radar images. *Neural Networks*, 12 :499–511, 1999.
- [Mink, 1996] Jonathan W. Mink. The basal ganglia : Focused selection and inhibition of competing motor programs. *Progress in Neurobiology*, 50(4) :381–425, 1996.
- [Mitchell *et al.*, 1993] Melanie Mitchell, Peter T. Hrabar, and James P. Crutchfield. Revisiting the edge of chaos : Evolving cellular automata to perform computations. *Complex Systems*, 7 :89–130, 1993.
- [Moran *et al.*, 2007] R.J. Moran, S.J. Kiebel, K.E. Stephan, R.B. Reilly, J. Daunizeau, and K.J. Friston. A neural mass model of spectral responses in electrophysiology. *NeuroImage*, 37 :706–720, 2007.
- [Mountcastle, 1957] V. Mountcastle. Modality and topographic preperities of single neurons in cat's somatic sensory cortex. *Journal of Neurophysiology*, 20 :408–434, 1957.
- [Mountcastle, 1997] Vernon B. Mountcastle. The columnar organization of the neocortex. *Brain*, 120 :701–722, 1997.
- [Murdock *et al.*, 2006] J. Angela Murdock, Fernanda Botelho, and James E. Jamison. Persistence of spatial patterns produced by neural field equations. *Physica D*, 215 :106–116, 2006.
- [Myin and O'Regan, 2006] Erik Myin and J. Kevin O'Regan. Situated perception and sensation in vision and other modalities : A sensorimotor approach. In Phillips Robins and Murat Aydede, editors, *Cambridge Handbook of Situated Cognition*, chapter 14. Cambridge University Press, 2006.
- [NCS, 2008] NCS, 2008. <http://brain.unr.edu/ncsDocs>, Consulté le 15 janvier 2008.
- [Néda *et al.*, 1996] Z. Néda, B. Bakó, and E. Reesa. The dripping faucet revisited. *Chaos*, 6(1), 1996.
- [NEST, 2008] NEST, 2008. http://www.nest-initiative.uni-freiburg.de/index.php/Main_Page, Consulté le 15 janvier 2008.
- [NeuroComp, 2006] NeuroComp, 2006. <http://neurocomp.risc.cnrs.fr>.
- [NEURON, 2008] NEURON, 2008. <http://neuron.duke.edu>, Consulté le 4 janvier 2008.
- [Noë, 2002] Alva Noë. Is the visual world a grand illusion? *Journal of Consciousness Studies*, 9(5-6), 2002.
- [Orban *et al.*, 2004] Guy A. Orban, David Van Essen, and Wim Vanduffel. Comparative mapping of higher visual areas in monkeys and humans. *Trends in Cognitive Sciences*, 8(7) :315–324, 2004.

-
- [O'Regan and Noë, 2001] J. K. O'Regan and A. Noë. A sensorimotor account of vision and visual consciousness. *Behavioral and Brain Sciences*, 24(5) :939–1011, 2001.
- [Oster and Bressloff, 2006] Andrew M. Oster and Paul C. Bressloff. A developmental model of ocular dominance column formation on a growing cortex. *Bulletin of Mathematical Biology*, 68(1) :73–98, 2006.
- [Packard, 1988] N. H. Packard. Adaptation toward the edge of chaos. In J. A. S. Kelso, A. J. Mandell, and M. F. Shlesinger, editors, *Dynamic Patterns in Complex Systems*, pages 293–301. World Scientific, 1988.
- [parXXL, 2010] parXXL, 2010. <https://gforge.inria.fr/projects/parxxl/>.
- [Paugam-Moisy *et al.*, 2007] Hélène Paugam-Moisy, Régis Martinez, and Samy Bengio. A supervised learning approach based on stdp and polychronization in spiking neuron networks. In *European Symposium on Artificial Neural Networks (ESANN'2007)*, 2007.
- [Pearl, 2000] Judea Pearl. *Causality : Models, Reasoning, and Inference*. Cambridge University Press, 2000.
- [Pietquin *et al.*, 2011] Olivier Pietquin, Matthieu Geist, Senthilkumar Chandramohan, and Hervé Frezza-Buet. Sample-Efficient Batch Reinforcement Learning for Dialogue Management Optimization. *ACM Transactions on Speech and Language Processing*, 7(3), May 2011. 21 pages.
- [Pinto and Ermentrout, 2001a] D.J. Pinto and G.B. Ermentrout. Spatially structured activity in synaptically coupled neuronal networks : I. traveling fronts and pulses. *SIAM J. Appl. Math.*, 62 :206–225, 2001.
- [Pinto and Ermentrout, 2001b] D.J. Pinto and G.B. Ermentrout. Spatially structured activity in synaptically coupled neuronal networks : II. lateral inhibition and standing pulses. *SIAM J. Appl. Math.*, 62 :226–243, 2001.
- [Pollard *et al.*, 2006] Katherine S. Pollard, Sofie R. Salama, Nelle Lambert, Marie-Alexandra Lambot, Sandra Coppens, Jakob S. Pedersen, Sol Katzman, Bryan King, Courtney Onodera, Adam Siepel, Andrew D. Kern, Colette Dehay, Haller Igel, Manuel Ares, Pierre Vanderhaeghen, and David Haussler. An rna gene expressed during cortical development evolved rapidly in humans. *Nature*, 443 :167–172, 2006.
- [Premti, 1983] Frederik Premti. *Méthodes Stochastiques dans les problèmes de Placement*. PhD thesis, Université Joseph Fourier, Grenoble, 1983.
- [Pulvermüller, 2003] Friedemann Pulvermüller. *The Neuroscience of Language*. Cambridge University Press, 2003.
- [Rao and Ballard, 1999] Rajesh P. N. Rao and Dana H. Ballard. Predictive coding in the visual cortex : a functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, 2(1) :79–87, 1999.
- [Rendell, 2002] Paul Rendell. Turing universality of the game of life. *Collision-based computing*, pages 513–539, 2002.
- [Rice *et al.*, 2007] Kenneth L. Rice, Christopher N. Vutsinas, and Tarek M. Taha. A preliminary investigation of a neocortex model implementation on the cray xd1. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC07)*, 2007.
- [Ritter *et al.*, 1992] Helge Ritter, Thomas Martinetz, and Klaus Schulten. *Neural Computation and Self-Organizing Maps - An Introduction*. Addison-Wesley, 1992.

- [Robert, 1995] François Robert. *Les Systèmes dynamiques discrets*. Springer, 1995.
- [Rochel and Martinez, 2003] Olivier Rochel and Dominique Martinez. An event-driven framework for the simulation of networks of spiking neurons. In *Proc. 11th European Symposium on Artificial Neural Networks, ESANN 2003*, 2003.
- [Roska *et al.*, 1995] Tamás Roska, Leon O. Chua, Dietrich Wolf, Tibor Kozek, Ronald Tetzlaff, and Frank Puffer. Simulating nonlinear waves and partial differential equations via cnn-part i : Basic techniques. *IEEE Transactions on Circuits and Systems-I : Fundamental theory and applications*, 42(10) :807–815, october 1995.
- [Rougier and O’Reilly, 2002] N.P. Rougier and R. C. O’Reilly. Learning representations in a gated prefrontal cortex model of dynamic task switching. *Cognitive Science*, 26 :503–520, 2002.
- [Rougier and Vitay, 2006] Nicolas P. Rougier and Julien Vitay. Emergence of attention within a neural population. *Neural Networks*, 19(5) :573–581, 2006.
- [Rougier *et al.*, 2005] N.P. Rougier, D. Noelle, T. S. Braver, J. D. Cohen, and R. C. O’Reilly. Prefrontal cortex and the flexibility of cognitive control : Rules without symbols. *Proceedings of the National Academy of Sciences*, 102 :7338–7343, 2005.
- [Rougier, 2000] Nicolas Rougier. *Modèles de mémoires pour la navigation autonome*. PhD thesis, Université Henri-Poincaré, Nancy 1, 2000.
- [Rougier, 2001] Nicolas Rougier. Hippocampal auto-associative memory. In *International Joint Conference on Neural Networks (IJCNN 01)*, 2001.
- [Rougier, 2006] Nicolas P. Rougier. Dynamic neural field with local inhibition. *Biological Cybernetics*, 94(3) :169–179, 2006.
- [Sabatini *et al.*, 2004] Silvio P. Sabatini, Fabio Solari, and Luca Secchi. A continuum-field model of visual cortex stimulus-driven behaviour : emergent oscillations and coherence fields. *Neurocomputing*, 57 :411–433, 2004.
- [Sabatini *et al.*, 2005] Silvio P. Sabatini, Fabio Solari, and Luca Secchi. Emergence of oscillations and spatio-temporal coherence states in a continuum-model of excitatory and inhibitory neurons. *BioSystems*, 79 :101–108, 2005.
- [Sargeni and Bonaiuto, 2005] Fausto Sargeni and Vincenzo Bonaiuto. Programmable cnn analogue chip for rd-pde multi-method simulation. *Analog Integrated Circuits and Signal Processing*, 44 :283–292, 2005.
- [Schneider and Erlhagen, 2002] Stefan Schneider and Wolfram Erlhagen. A neural field model for saccade planning in the superior colliculus : speed-accuracy tradeoff in the double-target paradigm. *Neurocomputing*, 44-46 :623–628, 2002.
- [Schultz, 2001] Wolfram Schultz. Reward signaling by dopamine neurons. *The Neuroscientist*, 7(4) :293–302, 2001.
- [Schwabe *et al.*, 2006] Lars Schwabe, Klaus Obermayer, Alessandra Angelucci, and Paul C. Bressloff. The role of feedback in shaping the extra-classical receptive field of cortical neurons : A recurrent network model. *The Journal of Neuroscience*, 26(36) :9117–9129, 2006.
- [Searle, 1980] John Searle. Minds, brains, and programs. *Behavioral and Brain Sciences*, 3 :417–424, 1980.
- [Seth *et al.*, 2004] Anil K. Seth, Jeffrey L. McKinstry, Gerald M. Edelman, and Jeffrey L. Krichmar. Visual binding through reentrant connectivity and dynamic synchronization in a brain-based device. *Cerebral Cortex*, 14(11) :1185–1199, 2004.

-
- [Shawe-Taylor and Cristianini, 2000] John Shawe-Taylor and Nello Cristianini. *Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000.
- [Shevade *et al.*, 1999] S. K. Shevade, S. S. Keerthi, C. Bhattacharyya, and K. R. K. Murthy. Improvements to smo algorithm for svm regression. Technical Report CD-99-16, National University of Singapore, 1999.
- [Simmering *et al.*, 2007] Vanessa R. Simmering, Anne R. Schutte, and John P. Spencer. Generalizing the dynamic field theory of spatial cognition across real and developmental time scales. *Brain Research*, 2007. in press.
- [Slavova and Mladenov, 2004] Angela Slavova and Valeri Mladenov, editors. *Cellular Neural Network : Theory and Applications*. Nova Science Publishers, New York, 2004.
- [Slavova and Zecca, 2003] Angela Slavova and Pietro Zecca. Cnn model for studying dynamics and travelling wave solutions of fitzhugh-nagumo equation. *Journal of Computational and Applied Mathematics*, 151(1) :12–24, 2003.
- [Slavova and Zecca, 2007] Angela Slavova and Pietro Zecca. Complex behavior of polynomial fitzhugh-nagumo cellular neural network model. *Nonlinear Analysis : Real World Applications*, 8(4) :1331–1340, 2007.
- [Slavova, 2000] Angela Slavova. Application of some mathematical methods in the analysis of cellular neural networks. *Journal of Computational and Applied Mathematics*, 114 :387–404, 2000.
- [Slavova, 2003] Angela Slavova. *Cellular neural networks : dynamics and modelling*. Kluwer Academic Publishers, Boston, 2003.
- [Slocum *et al.*, 2000] Andrew C. Slocum, Douglas C. Downey, and Randall D. Beer. Further experiments in the evolution of minimally cognitive behavior : From perceiving affordances to selective attention. In *Sixth International Conference on Simulation of Adaptive Behavior*, 2000.
- [Soula *et al.*, 2005] Hédi Soula, Aravind Alwan, and Guillaume Beslon. Learning at the edge of chaos : Temporal coupling of spiking neuron controller for autonomous robotics. In *AAAI Spring symposia on developmental robotics*, 2005.
- [Sporns, 2006] Olaf Sporns. Small-world connectivity, motif composition, and complexity of fractal neuronal connections. *Biosystems*, 85(1) :55–64, 2006.
- [Stavrinou *et al.*, 2007] Maria L. Stavrinou, Stefania Della Penna, Vittorio Pizzella, Kathya Torquati, Francesco Cianflone, Raffaella Franciotti, Anastasios Bezerianos, Gian Luca Romani, and Paolo Maria Rossini. Temporal dynamics of plastic changes in human primary somatosensory cortex after finger webbing. *Cerebral Cortex*, 17(9) :2134–2142, 2007.
- [Sur and Rubenstein, 2005] Mriganka Sur and John L. R. Rubenstein. Patterning and plasticity of the cerebral cortex. *Science*, 310(5749) :805–810, 2005.
- [Sutton and Barto, 1981] Richard S. Sutton and Andrew G. Barto. Toward a modern theory of adaptive network : Expectation and prediction. *Psychological Review*, 88(2) :135–170, 1981.
- [Sutton and Barto, 1998] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning : An Introduction*. MIT Press, Cambridge, MA, 1998.
- [Taylor, 1997] John G. Taylor. Neural networks for consciousness. *Neural Networks*, 10(7) :1207–1225, 1997.
- [Thorpe and Gautrais, 1998] Simon J. Thorpe and Jacques Gautrais. Rank order coding. In J. Bower, editor, *Computational Neuroscience*, Trends in Research, pages 113–118. Plenum Press, New York, 1998.

- [Thorpe *et al.*, 01] S. Thorpe, A. Delorme, and R. Van Rullen. Spike-based strategies for rapid processing. *Neural Networks*, 14 :715–725, 01.
- [Tian and Pearl, 2001] J. Tian and J. Pearl. Causal discovery from changes : a bayesian approach. In *Proceedings of UAI 17*, pages 512–521, 2001.
- [Torres-Huitzil *et al.*, 2005] César Torres-Huitzil, Bernard Girau, and Claudio Castellanos-Sánchez. On-chip visual perception of motion : A bio-inspired connectionist model on fpga. *Neural Networks*, 18 :557–565, 2005.
- [Toussaint, 2006] Marc Toussaint. A sensorimotor map : Modulating lateral interactions for anticipation and planning. *Neural Computation*, 18 :1132–1155, 2006.
- [Troyer *et al.*, 1998] Todd W. Troyer, Anton E. Krukowski, Nicholas J. Priebe, and Kenneth D. Miller. Contrast-invariant orientation tuning in cat visual cortex : Thalamocortical input tuning and correlation-based intracortical connectivity. *The Journal of Neuroscience*, 18(15) :5908–5927, 1998.
- [Turing, 1936] Alan. M. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proc. London Math. Soc.*, 2(42) :230–265, 1936. See also the correction, on number 43 :544–546.
- [Turing, 1950] Alan. M. Turing. Computing machinery and intelligence. *Mind*, 49 :433–460, 1950.
- [van Hemmen and Kempter, 1998] J. L. van Hemmen and R. Kempter. Hebbian learning of temporal correlations : Sound localization in the barn owl auditory system. In D. Beysens and G. Forgacs, editors, *Dynamical Networks in Physics and Biology*, pages 285–296. Springer, 1998.
- [Vapnik, 2000] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Statistics for Engineering and Information Science. Springer, 2000.
- [Venkov *et al.*, 2007] N. A. Venkov, S. Coombes, and P.C. Matthews. Dynamic instabilities in scalar neural field equations with space-dependent delays. *Physica D*, 232 :1–15, 2007.
- [Vialle *et al.*, 2005] Stephane Vialle, Amelia De Vivo, and Fabrice Sabatier. A grid architecture for comfortable robot control. In P.M.A.Sloot, A.G. Hoekstra, T. Priol, A. Reinefeld, and M. Bubak, editors, *Advances in Grid Computing - EGC 2005 : European Grid Conference, Amsterdam, The Netherlands, February 14-16, 2005, Revised Selected Papers*, volume 3470 of *Lecture Notes in Computer Science*, pages 344–353. Springer, 2005.
- [Vialle, 2002] Stéphane Vialle. Parallélisation de systèmes de calculs distribués d’orientation cellulaire, sur architecture mimd. Technical report, Université Henri Poincaré - Nancy I / Supélec, 2002. M’emoire d’Habilitation à Diriger les Recherches.
- [Watts and Strogatz, 1998] D. J. Watts and S. H. Strogatz. Collective dynamics of ’small-world’ networks. *Nature*, 393 :440–442, 1998.
- [Weidig, 2005] Jacques Weidig. 2005. Technical report, Conservatoire National des Arts et Métiers (CNAM), 2005. Mémoire de fin d’études.
- [Wiemer, 2003] Jan C. Wiemer. The time-organized map algorithm : Extending the self-organizing map to spatiotemporal signals. *Neural Computation*, 15 :1143–1171, 2003.
- [Willshaw and Malsburg, 1976] D. J. Willshaw and C. Von Der Malsburg. How patterned neural connections can be set up by self-organization. *Proceedings of the Royal Society of London*, 194(1117) :431–445, 1976. Series B, Biological Sciences.

-
- [Wolfram, 1984] Stephen Wolfram. Universality and complexity in cellular automata. *Physica D*, 10 :1–35, 1984.
- [XPP-auth, 2008] XPP-auth, 2008. <http://www.math.pitt.edu/~bard/xpp/xpp.html>, Consulté le 4 janvier 2008.
- [Zhu and von der Malsburg, 2004] Junmei Zhu and Christoph von der Malsburg. Maplets for correspondence-based object recognition. *Neural Networks*, 17(8-9) :1311–1326, 2004.

Index

- émergence, 29
- énaction, 9

- activation multimodale cohérente, 105
- affordance, 20
- apprentissage par renforcement, 52, 67, 125, 154
- approche située, 8, 10
- arbre d'appel, 124
- architecture, 43
- architecture corticale, 77, 87
- attracteur étrange, 38
- attracteur de Lorenz, 38
- auto-organisation, 107
- auto-organisation conjointe, 109
- automates cellulaires, 42
- autopoïèse, 9

- balbutiement moteur, 126
- bassin d'attraction, 38
- bifurcations, 39
- bijama, 102
- bistable, 121, 124
- breathers, 58

- calcul à grain fin, 155
- calcul cortical, 33, 95, 102
- Cellular Neural Networks, 45, 138
- cellules de lieux, 17
- chambre chinoise, 1
- champ de vecteurs, 38
- champ récepteur, 42, 79, 80, 104
- champs de neurones dynamiques, 53
- Change Blindness, 20
- chaos déterministe, 38
- codage monotone, 84
- codage par rang, 70
- codage prédictif, 89
- codage tabulaire, 84, 95, 111, 128, 137, 153, 157
- cogito, 3
- colonne corticale, 72
- compétition, 77, 82, 96, 103, 121

- connection stripes, 85
- connexions typées, 75, 98

- Echo State Machine, 50
- Edge of chaos, 49
- Effet Baldwin, 65
- exposants de Lyapunov, 39

- fading memory, 49
- fonction de Lyapunov, 40
- frame problem, 8

- grabiness, 20
- gradients de modalités, 87
- Grande Illusion, 17
- graphes d'itérations, 44
- Growing Neural Gas, 140
- grumpf, 97

- hiérarchie, 87

- IA forte, 1
- indices primaires, 78, 104
- indices secondaires, 78, 111
- instabilités de Turing, 56
- integrate and fire, 48

- Jeu de la vie, 43

- latence, 31
- Liquid computing, 49
- Liquid State Machine, 50
- localiste, 96

- mémoire épisodique, 189
- mémoire adressable par le contenu, 53
- mémoire provisoire, 12, 21, 124
- maxicolonne corticale, 72, 78
- microcolonne corticale, 72
- minicolonne corticale, 72
- mise à jour, 42
- mise à jour asynchrone, 121

- mises à jour asynchrones, 44
- mises à jour synchrones, 44
- modèle génératif, 89
- modèles compartimentaux, 46, 68
- modèles de neurones événementiels, 48
- Motor Pattern Generator, 85

- neural mass, 47
- neurone grand-mère, 96
- neurones miroir, 156

- on-center/off-surround, 55
- opérateurs de Gauss-Seidel, 44
- orbite, 37

- perception amodale, 20
- poids latéraux, 55
- point de capiton, 122
- point fixe, 38
- polychronie, 66
- potentiels d'actions, 48
- problème du binding, 94
- programmation génétique, 65
- proprioception, 126
- prototype, 82

- récompense, 124
- réseau de Hopfield, 53
- réseaux petit-monde, 32
- résonance, 88
- relaxation, 52, 88, 121
- représentation, 9, 96
- reservoir, 50
- Reservoir computing, 49
- ressources de calcul extensives, 34, 75
- restitution de cohérence, 52, 57, 60, 77, 82, 88, 105, 128, 132
- return map, 42

- sérialiser, 21, 77, 95, 121, 124
- sciences cognitives, 1
- Self-Organized Criticality, 49
- simulateurs événementiels, 68
- Spectral Timing, 51
- spike, 48
- Spike-time-dependent Plasticity, 66
- SRM0, 48
- STDP, 66
- subsumption architecture, 16, 117

- symmetry breaking, 47
- système dynamique, 37
- systèmes complexes, 29
- systèmes de grammaires, 29

- temps logique, 121, 124, 185, 189
- temps physique, 120, 122, 185
- topographique, 80, 104
- trace, 120, 122

- Value Iteration, 52, 125